Highly Parallel Autoregressive Entity Linking with Discriminative Correction

Nicola De Cao [1,2], Wilker Aziz [1], Ivan Titov [1,2]
[1]University of Amsterdam, [2]University of Edinburgh
{ nicola.decao, w.aziz, titov } @uva.nl

Abstract

Generative approaches have been recently shown to be effective for both Entity Disambiguation and Entity Linking (*i.e.*, joint mention detection and disambiguation). However, the previously proposed autoregressive formulation for EL suffers from i) high computational cost due to a complex (deep) decoder, ii) non-parallelizable decoding that scales with the source sequence length, and iii) the need for training on a large amount of data. In this work, we propose a very efficient approach that parallelizes autoregressive linking across all potential mentions and relies on a shallow and efficient decoder. Moreover, we augment the generative objective with an extra discriminative component, *i.e.* a correction term which lets us directly optimize the generator's ranking. When taken together, these techniques tackle all the above issues: our model is >70 times faster and more accurate than the previous generative method, outperforming state-of-the-art approaches on the standard English dataset AIDA-CoNLL. [1]

1 Introduction

Entity Linking (EL; Bunescu and Paşca, 2006; Cucerzan, 2007; Dredze et al., 2010; Hoffart et al., 2011; Le and Titov, 2018) is a fundamental task in NLP employed as a building block for text understanding (Févry et al., 2020a; Verga et al., 2020), question answering (Nie et al., 2019; Asai et al., 2020; De Cao et al., 2019), dialog modeling (Dinan et al., 2019; Sevegnani et al., 2021), and information extraction (Sarawagi, 2008; Martinez-Rodriguez et al., 2020), to name a few. Popular earlier methods address the Mention Detection (MD) and Entity Disambiguation (ED) stages of EL separately (Ceccarelli et al., 2013; Daiber et al., 2013; Steinmetz and Sack, 2013; Piccinno and Ferragina,

2014) while modern techniques leverage their mutual dependency (Kolitsas et al., 2018; Broscheit, 2019; Martins et al., 2019). A new line of work (De Cao et al., 2021a,b) departs from linking mentions using a vector space and instead uses large language models fine-tuned with a generative objective (*i.e.*, predicting a textual identifier as the entity identifier).

Employing autoregressive language models better leverages the implicit knowledge accumulated during pre-training, exploiting a full cross-encoder of entities and their context. For ED, autoregressive generation is remarkably good (even in multilingual settings), while for EL, although state-of-the-art on multiple datasets, it suffers from several and critical limitations. The generative model of De Cao et al. (2021a) outputs a version of the input document which is markup-annotated with mentions linked to their respective entities. This necessitates using an autoregressive decoder, precluding parallelism across mentions. Generation also has a high computational cost due to relying on a complex and deep Transformer (Vaswani et al., 2017) decoder. Transformers are state-less and their memory footprint scales with sequence length, making them memory-consuming when generating long sequences. Additionally, Transformers-based decoders are notably data-hungry, and their effective training requires large amounts of data. For example, De Cao et al. (2021a) had to pre-train their model on Wikipedia abstracts.

In this work, we revisit the generative approach to EL and generate mention-entity pairs conditionally independently given the input. This allows for parallelism across mentions, which we exploit by employing a shallow LSTM-based decoder. To optimize more explicitly the generator's ranking, we use a discriminative correction term that pushes the score of the correct predictions to be higher than the rest. Moreover, to enable conditioning on long inputs, we employ an efficient Transformer

---

[1]Source code available at https://github.com/nicola-decao/efficient-autoregressive-EL
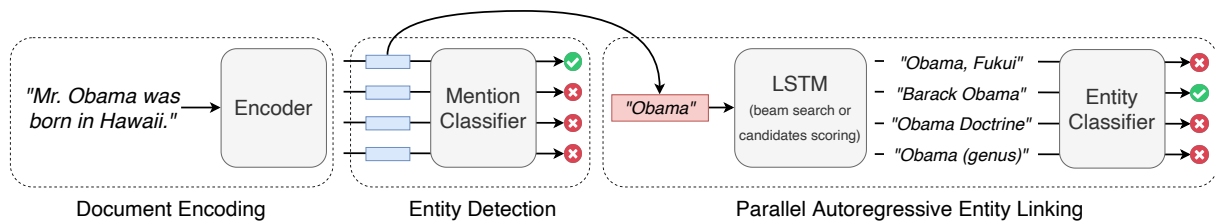
7662

Figure 1: Outline of our model: a Transformer-based *document encoder* embeds a document into vectors (the encoder is designed to support long text). Then, an *entity detection* module classifies which spans in the document are entity mentions. Conditioning on a mention embedding, an *entity linking* module first uses an LSTM to either generate or score candidates' textual identifiers and then a classifier to re-rank the candidates.

encoder (Beltagy et al., 2020) designed to support long sequences. Figure 1 outlines our model.

Contributions We propose a highly parallel model for autoregressive entity linking that retains the advantages of being generative while being >70 times faster than a previous generative formulation and as fast as non-generative models. We optimize for the correctness of the decoder's ranking with a discriminative loss to improve autoregressive EL further. The model outperforms state-of-the-art approaches on the standard English AIDA dataset.

2 Background

Task Entity Linking (EL) is the task of predicting a set $\mathcal{Y}$ of mention-entity pairs contained in some input text x (Hoffmann et al., 2011). Each mention m is a pair of start and end positions $\langle m_s, m_e \rangle$ indicating a span in x. Each mention m refers to an entity e in a fixed Knowledge Base (KB)—note that entities can be referred to with multiple ambiguous surface forms (*e.g.* in Wikidata "NYC" and "New York" both refers to the entity "New York City"[2]).

Related work EL is typically decomposed in Mention Detection (MD, *i.e.*, the task of finding mention spans in text) and Entity Disambiguation (ED, *i.e.*, the task of disambiguating a mention to its respective entity). Many methods (Hoffart et al., 2011; Piccinno and Ferragina, 2014; Steinmetz and Sack, 2013) treat these sub-tasks separately, training different modules. More modern approaches – known as end-to-end EL – instead use a shared (typically neural) architecture. Kolitsas et al. (2018) use a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) as an encoder and then local and global scoring functions to link mentions. They exploit pre-computed entity embeddings by Ganea and Hofmann (2017) and match the embeddings

to contextualized mention representations. Martins et al. (2019) also explore joint learning of Named Entity Recognition (NER) and EL showing that the two tasks benefit from joint training, while Li et al. (2020) approach EL specifically for questions.

In this work, we focus on monolingual EL in English while there is a line of work that explores cross-lingual entity linking (McNamee et al., 2011; Ji et al., 2015), that is linking from any source language to a standard one (*e.g.* English), and multilingual entity linking (Botha et al., 2020) that is a generalization of both.

Autoregressive Linking The GENRE model by De Cao et al. (2021a) departs from framing EL as matching in vector space, and instead frames it as a sequence-to-sequence problem. GENRE tackles MD and ED for all mention-entity pairs jointly by autoregressively generating a version of the input markup-annotated with the entities' unique identifiers expressed in natural language. Although we focus on EL, GENRE was also applied to ED alone as well as to page-level document retrieval for fact-checking, open-domain question answering, slot filling, and dialog (Petroni et al., 2021). mGENRE (De Cao et al., 2021b) is the multilingual extension of GENRE.

Modern techniques (Wu et al., 2020; Botha et al., 2020) are based on a dense retriever module that uses maximum inner-product search (MIPS) to match mention vectors to entity embeddings. In contrast with MIPS for linking, generative models i) exploit *knowledge* learned during pre-training, ii) are memory-efficient as they do not need to store pre-computed entity representations, and iii) are full cross-encoders of context and entity since decoders can use attention to context. Bi-encoders solutions may be sub-optimal and memory inefficient although memory-efficient dense retrieval has recently received attention (Izacard et al., 2020;

---

[2]https://www.wikidata.org/wiki/Q60

Min et al., 2021; Lewis et al., 2021).

A caveat of joint modeling all mention-entity pairs with an autoregressive model (*i.e.*, without any independence assumptions) is the lack of parallelism, which makes GENRE extremely slow for the complete task of EL. In addition, generation of open-ended text calls for a deep decoder and thus requires very large corpora for training.

3 Method

Our method learns by generating observed mention-entity pairs $\mathcal{Y}$ given an input document x. To enable, we assume that, given the document x, each mention-entity pair $\langle m, e \rangle \in \mathcal{Y}$ is independent of one another. Moreover, each pair's probability is further factorized as a product of an MD and an ED components: $p(\mathcal{Y}|x, \theta) \overset{\text{ind.}}{=}$

$$\prod_{\langle m,e \rangle \in \mathcal{Y}} p(m|x, \theta_{\text{MD}}) \, p(e|m, x, \theta_{\text{ED}}) \,, \quad (1)$$

where $\theta = \theta_{\text{MD}} \cup \theta_{\text{ED}}$ is a shared set of parameters (and $\theta_{\text{MD}} \cap \theta_{\text{ED}}$ need not be empty). To provide our models with a rich representation of the document, we encode it using a Longformer (Beltagy et al., 2020), a Transformer pre-trained with a masked language model objective that is designed to support long sequences.

Mention Detection   There are different ways to model $p(m|x, \theta_{\text{MD}})$ (*i.e.*, the probability that the span m in x contains a mention). One is to score all possible spans which requires a number of evaluations that is quadratic in sequence length. For long documents, that is clearly unfeasible. Thus, for maximizing efficiency, we opt for factorizing the probability of a span as the probability of its start m_s times the conditional probability of its end m_e given the start: $p(m|x, \theta_{\text{MD}}) =$

$$p(m_s|x, \theta_{\text{MD}}) \, p(m_e|m_s, x, \theta_{\text{MD}}) \,. \quad (2)$$

The first term is the probability that position m_s starts a mention, and the second is the probability that the mention has size $m_e - m_s + 1$, to which we give categorical treatment.[3] Such factorization allows both for fast training and inference. During training, mentions are known. For inference, we consider only the positions for which the probability of starting a mention exceeds a threshold chosen to maximise micro-F_1 on the validation set.

---

[3]We limit the maximum number of tokens per span to 15 to avoid memory overhead (in the training set there is no mention with more than 12 tokens).

Entity Disambiguation   The disambiguation module learns to generate the unique name of e autoregressively (token by token) from left to right:

$$p(e|x, m, \theta_{\text{ED}}) = \prod_{i=1}^{|t|} p(t_i|x, m, t_{<i}, \theta_{\text{ED}}) \,, \quad (3)$$

where t is the unique name of e in the KB. To fully exploit our design's potential for parallelism across mentions, we use a small single-layered LSTM (Hochreiter and Schmidhuber, 1997). This language model is not constrained to generating only valid entity names, besides, maximum likelihood training does not directly optimize for the correctness of the generator's ranking. To mitigate those issues, when training the architecture, we employ an auxiliary loss based on a discriminative classifier that assigns probability

$$p(e|x, m, \theta_{\text{ED}}) = \frac{\exp(f(x, m, t; \theta_{\text{ED}}))}{\sum_c \exp(f(x, m, c; \theta_{\text{ED}}))} \,, \quad (4)$$

where f is an MLP (details in Section 4.2), t is the unique name of e and the normalization is over all entities in the KB (*i.e.*, their unique names).

Parameter Estimation   We estimate the parameters of all components jointly as to maximize the model's likelihood given a dataset of observations using stochastic gradient descent (SGD; Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952; Bottou, 2012). For the language model component, we employ length normalization (Sutskever et al., 2011, 2014) and label smoothing (Szegedy et al., 2016). All components are further regularized with dropout (Srivastava et al., 2014). The classification loss is the negative logarithm of Equation 4, and we approximate the normalization constant via negative sampling, with samples drawn from a candidate set specific to each training instance.

4 Experiments

4.1 Setting

We use the standard English AIDA-CoNLL splits (Hoffart et al., 2011) for training, validation (*i.e.*, for doing model selection), and test. See Table 1 for statistics of this dataset. AIDA provides full supervision for both MD and ED. We only link mentions that have a valid gold KB entity, a setting referred to as *InKB* evaluation (Röder et al., 2018). This is in line with many previous models (Luo

Split	Documents	Mentions
Training	942	18,540
Validation	216	4,791
Test	230	4,485

Table 1: Statistics of the AIDA-CoNLL standard splits (Hoffart et al., 2011) dataset.

et al., 2015; Ganea and Hofmann, 2017; Yamada et al., 2016) and all systems we compare to. As in several previous approaches, for linking we assume the availability of a pre-computed set of candidates instead of considering the whole KB. For that, we use the candidates by Pershina et al. (2015). We also use these candidates to provide negative samples for the discriminative loss during training (see Equation 4).

4.2 Architecture details

As the document encoder, we use a Longformer (Beltagy et al., 2020). A Longformer is a RoBERTa (Liu et al., 2019) model with a limited attention window (we use 128 tokens). It has 12 layers, of which we use the first 8 (for faster computation), a hidden size of 768, 12 heads, for a total of 149M parameters. The MD modules (*i.e.*, $p(m_s|x, \theta_{\text{MD}})$ and $p(m_e|x, m_s, \theta_{\text{MD}})$) are both implemented as feed forward NNs that take as inputs contextualized token embeddings. They have architecture: [LayerNorm, 128, ReLU, LayerNorm, 1]. We applied dropout of 0.1 before linear projections. The autoregressive ED module $p(t_i|m, x, t_{<i}, \theta_{\text{ED}})$ is implemented with an LSTM. Three feed-forward NNs predict the first hidden state, the first context vector, and a vector to append to each decoding step. The predictions are a function of the start and end embeddings of a mention. All 3 FFNNs have architecture [LayerNorm, 768, ReLU, LayerNorm, 768]. The LSTM has an input size of 1536 and a hidden size of 768. The LSTM uses the shared input embedding from the Longformer encoder and an output head initialized from the Longformer. The discriminative classifier $f(x, m, t; \theta_{\text{ED}}))$ is a feed-forward NN that takes as an input a vector representation of a mention and the last context vector of the LSTM. The FFNN has architecture [LayerNorm, 768, ReLU, LayerNorm, 1]. Our whole model has a total of 202M parameters. We manually employ search from using Layer normalization (Ba et al., 2016) or not and the number of the Longformer layers to use.

Method	Micro-F_1
Hoffart et al. (2011)	72.8
Steinmetz and Sack (2013)	42.3
Daiber et al. (2013)	57.8
Moro et al. (2014)	48.5
Piccinno and Ferragina (2014)	73.0
Kolitsas et al. (2018)	82.4
Peters et al. (2019)	73.7
Broscheit (2019)	79.3
Martins et al. (2019)	81.9
van Hulst et al. (2020)[†]	80.5
Févry et al. (2020b)	76.7
De Cao et al. (2021a)	<u>83.7</u>
Kannan Ravi et al. (2021)	83.1
Ours	**85.5**
Ablations (ours)	
LM score only	81.5
Classifier score only	81.7
Beam Search w/ candidates	84.9
Beam Search w/o candidates	49.4*

Table 2: Results (InKB) on the AIDA test set and some ablation of our system. **Bold** indicates best model and <u>underline</u> indicates previous state-of-the-art. [†]Results from the Wikipedia 2019 setting as opposed to the 2014 setting (older dump and fewer entities). *Our generative component has only seen a fraction of entities identifiers ($\approx$2k compared to the KB size of $\approx$500k).

4.3 Training details

We optimize our model employing Adam (Kingma and Ba, 2015) with weight decay of 1e-2. We use a learning rate of 1e-4 for the Longformer and a learning rate of 1e-3 for all other components. We use a learning rate linear decay schedule for a maximum of 10,000 steps with 500 warm-up steps. We train with a batch size of 32 for a maximum of 100 epochs, and we do model selection on micro-F_1 on the validation set. We also optimized the threshold for the MD component with a grid search between -5 and 5 with steps 0.1 measuring micro-F_1 on the validation set. Training takes approximately one hour on 4 GPUs Nvidia Titan X 12 GB.

4.4 Results

Table 2 summarizes the main results of this work. Our method reduces the micro-F_1 error from the previous state-of-the-art method by 11%. The EL score can be also decomposed in Mention Detec-

Method	# Queries / Sec
Kolitsas et al. (2018)	$7.39_{\pm 5.03}$
De Cao et al. (2021a)	$0.12_{\pm 0.08}$
Ours	$\mathbf{8.69}_{\pm 5.13}$

Table 3: Inference speed of our model and the top-2 SOTA model from Table 2.

tion (MD) and Entity Disambiguation (ED) scores. Our method gets an MD micro-F_1 score of ≈ 94 and an ED micro-F_1 score of ≈ 92 (note that the EL task scores a prediction as correct when both mention detection and disambiguation are done correctly). Unfortunately, most of the baselines we compare to do not report this decomposition, and thus is difficult to systematically investigate where our method stands for MD and ED scores. Nevertheless, Kolitsas et al. (2018) is the second-best system in terms of EL micro-F_1, and the authors reported a ≈ 89 ED micro-F_1. As a comparison, Broscheit (2019) reported ≈ 88 and van Hulst et al. (2020) ≈ 84 ED micro-F_1. This suggests that our improvement mainly comes from improving ED.

Performance Evaluation In Table 3, we compare the speed of our system against the top-2 best baseline models from Table 2. We run 3 independent runs on the validation set and report the number of queries per second on GPU[4] feeding the models with one input at a time (*i.e.*, batch size of 1). For GENRE (De Cao et al., 2021a), we truncate sequences to the maximum supported length. Our model parallelizes the generation of all entity identifiers and dispenses with generating superfluous text (*i.e.*, the non-mentions) being >70 times faster than GENRE, which has to re-generate the whole source input left-to-right in order to fill in the mention-entity markup sequentially. Notably, our model is also slightly faster than Kolitsas et al. (2018) which is a well-established model for EL.

4.5 Analysis

We investigate the importance of different aspects of our model formulation in an ablation study. In Table 2 (bottom-half) we report all results.

Discriminative Correction We train with and without the discriminative correction term of Equation 4 to appreciate its impact in results. Using only

the LM component results in a 4% drop in performance: this is due to not optimizing directly for the correctness of the generator's ranking. Using the classifier alone also leads to a 4% drop in performance. Those ablations indicate that the auxiliary loss helps improve the generator's ranking.

Beam Search vs Complete Scoring To compare with previous work, we use pre-computed candidates for ED. This is feasible because the number of candidates to score is relatively small. However, in general, candidates might be too many and thus impractical to score them all. Thus, we test our model using Constrained Beam Search (CBS) as an approximation. When using CBS (with a beam size of 5), performance drops by <1%, and micro-F1 remains higher than that of every other baseline, demonstrating that our formulation is robust even in this setting.

Ablating Candidates One of the benefits of the generative formulation is the ability to generate entity names (autoregressively through CBS) without the need for candidates. Thus, we test our model using CBS without candidates (*i.e.*, all entities in the KB are viable candidates). In this setting, our model does not excel (42% drop in performance). The drop is not surprising: our generative component has only seen a fraction of entities identifiers (1537 out of $\approx 500,000$ in the KB). Indeed, previous methods (*e.g.*, De Cao et al. (2021a)) were pre-trained on the whole Wikipedia to mitigate this issue. We do not have the computational budget to do such pre-training so we leave this for follow-up work.

5 Conclusion

We revisit the generative approach to EL exploiting independence assumptions that enable parallelism across mentions with a shallow LSTM decoder. Despite a simple and scalable design, our model sets a new state-of-the-art on English AIDA without a large decoder pre-training.

[4]One Nvidia Titan X 12GB.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *Advances in NIPS 2016 Deep Learning Symposium*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China. Association for Computational Linguistics.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Dexter: an open source framework for entity linking. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval*, pages 17–20.

Silviu Cucerzan. 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)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021a. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021b. Multilingual Autoregressive Entity Linking. *ArXiv*, abs/2103.12528.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, Beijing, China. Coling 2010 Organizing Committee.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020a. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online. Association for Computational Linguistics.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020b. Empirical evaluation of pretraining strategies for supervised entity linking. In *Automated Knowledge Base Construction (AKAB)*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. A Memory Efficient Baseline for Open Domain Question Answering. *ArXiv*, abs/2012.15156.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *TAC*.

Manoj Prabhakar Kannan Ravi, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Jens Lehmann. 2021. CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 504–514, Online. Association for Computational Linguistics.

Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Phong Le and Ivan Titov. 2018. 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. Association for Computational Linguistics.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *ArXiv preprint*, abs/2102.07033.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal. Association for Computational Linguistics.

Jose L Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. 2020. Information extraction meets the semantic web: a survey. *Semantic Web*, pages 1–81.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2019. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196, Florence, Italy. Association for Computational Linguistics.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas Oard, and David Doermann. 2011. Cross-language entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 255–263, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned. *ArXiv*, abs/2101.00133.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2553–2566, Hong Kong, China. Association for Computational Linguistics.

Maria Pershina, Yifan He, and Ralph Grishman. 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*, pages 238–243, Denver, Colorado. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Francesco Piccinno and Paolo Ferragina. 2014. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625.

Sunita Sarawagi. 2008. *Information extraction*. Now Publishers Inc.

Karin Sevegnani, David M. Howcroft, Ioannis Konstas, and Verena Rieser. 2021. OTTers: One-turn topic transitions for open-domain dialogue. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2492–2504, Online. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Nadine Steinmetz and Harald Sack. 2013. Semantic multimedia information retrieval based on contextual descriptions. In *The Semantic Web: Semantics and Big Data*, pages 382–396, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1017–1024. Omnipress.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: an entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2197–2200. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *ArXiv preprint*, abs/2007.00849.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany. Association for Computational Linguistics.