Global Entity Disambiguation with Pretrained Contextualized Embeddings of Words and Entities

Ikuya Yamada[1,4] **Koki Washio**[2,4] **Hiroyuki Shindo**[3,4] **Yuji Matsumoto**[4]

ikuya@ousia.jp kwashio@g.ecc.u-tokyo.ac.jp shindo@is.naist.jp yuji.matsumoto@riken.jp

[1]Studio Ousia, Tokyo, Japan [2]The University of Tokyo, Tokyo, Japan
[3]Nara Institute of Science and Technology, Nara, Japan [4]RIKEN AIP, Tokyo, Japan

Abstract

We propose a new global entity disambiguation (ED) model based on contextualized embeddings of words and entities. Our model is based on a bidirectional transformer encoder (i.e., BERT) and produces contextualized embeddings for words and entities in the input text. The model is trained using a new *masked entity prediction* task that aims to train the model by predicting randomly masked entities in entity-annotated texts obtained from Wikipedia. We further extend the model by solving ED as a sequential decision task to capture global contextual information. We evaluate our model using six standard ED datasets and achieve new state-of-the-art results on all but one dataset.

1 Introduction

Entity disambiguation (ED) refers to the task of assigning entity mentions in a text to corresponding entries in a knowledge base (KB). This task is challenging because of the ambiguity between entity names (e.g., "World Cup") and the entities they refer to (e.g., `FIFA World Cup` or `Rugby World Cup`). Recent ED models typically rely on two types of contextual information: *local* information based on words that co-occur with the mention, and *global* information based on document-level coherence of the disambiguation decisions. A key to improve the performance of ED is to combine both local and global information as observed in most recent ED models.

In this study, we propose a novel ED model based on contextualized embeddings of words and entities. The proposed model is based on BERT (Devlin et al., 2019). Our model takes words and entities in the input document, and produces a contextualized embedding for each word and entity. Inspired by the masked language model (MLM) adopted in BERT, we propose *masked entity prediction* (MEP), a novel task that aims to train

the model by predicting randomly masked entities based on words and non-masked entities. We train the model using texts and their entity annotations retrieved from Wikipedia.

Furthermore, we introduce a simple extension to the inference step of the model to capture global contextual information. Specifically, similar to the approach used in past work (Fang et al., 2019; Yang et al., 2019), we address ED as a sequential decision task that disambiguates mentions one by one, and uses words and already disambiguated entities to disambiguate new mentions.

We evaluate the proposed model using six standard ED datasets and achieve new state-of-the-art results on all but one dataset. Furthermore, we will publicize our code and trained embeddings.

2 Background and Related Work

Neural network-based approaches have recently achieved strong results on ED (Ganea and Hofmann, 2017; Yamada et al., 2017; Le and Titov, 2018; Cao et al., 2018; Le and Titov, 2019; Yang et al., 2019). These approaches are typically based on embeddings of words and entities trained using a large KB (e.g., Wikipedia). Such embeddings enable us to design ED models that capture the contextual information required to address ED. These embeddings are typically based on conventional word embedding models (e.g., skip-gram (Mikolov et al., 2013)) that assign a fixed embedding to each word and entity (Yamada et al., 2016; Cao et al., 2017; Ganea and Hofmann, 2017).

Shahbazi et al. (2019) and Broscheit (2019) proposed ED models based on contextualized word embeddings, namely, ELMo (Peters et al., 2018) and BERT, respectively. These models predict the referent entity of a mention using the contextualized embeddings of the constituent or surrounding words of the mention. However, unlike our proposed model, these models address the task based only on local contextual information.
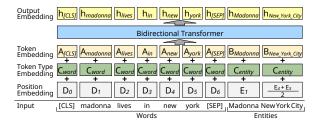
Figure 1: Architecture of the proposed contextualized embeddings of words and entities.

3 Contextualized Embeddings of Words and Entities for ED

Figure 1 illustrates the architecture of our contextualized embeddings of words and entities. Our model adopts a multi-layer bidirectional transformer encoder (Vaswani et al., 2017).

Given a document, the model first constructs a sequence of tokens consisting of words in the document and entities appearing in the document. Then, the model represents the sequence as a sequence of input embeddings, one for each token, and generates a contextualized output embedding for each token. Both the input and output embeddings have H dimensions. Hereafter, we denote the number of words and that of entities in the vocabulary of our model by V_w and V_e, respectively.

3.1 Input Representation

Similar to the approach adopted in BERT (Devlin et al., 2019), the input representation of a given token (word or entity) is constructed by summing the following three embeddings of H dimensions:

- **Token embedding** is the embedding of the corresponding token. The matrices of the word and entity token embeddings are represented as $\mathbf{A} \in \mathbb{R}^{V_w \times H}$ and $\mathbf{B} \in \mathbb{R}^{V_e \times H}$, respectively.

- **Token type embedding** represents the type of token, namely, word type (denoted by $\mathbf{C}_{word}$) or entity type (denoted by $\mathbf{C}_{entity}$).

- **Position embedding** represents the position of the token in a word sequence. A word and an entity appearing at the i-th position in the sequence are represented as $\mathbf{D}_i$ and $\mathbf{E}_i$, respectively. If an entity name contains multiple words, its position embedding is computed by averaging the embeddings of the corresponding positions.

Following BERT (Devlin et al., 2019), we insert special tokens [CLS] and [SEP] to the word sequence as the first and last words, respectively.

3.2 Masked Entity Prediction

To train the model, we propose *masked entity prediction* (MEP), a novel task based on MLM. In particular, some percentage of the input entities are masked at random; then, the model learns to predict masked entities based on words and non-masked entities. We represent masked entities using the special [MASK] entity token.

We adopt a model equivalent to the one used to predict words in MLM. Specifically, we predict the original entity corresponding to a masked entity by applying the softmax function over all entities in our vocabulary:

$$\hat{\mathbf{y}}_{MEP} = \text{softmax}(\mathbf{Bm} + \mathbf{b}_o), \quad (1)$$

where $\mathbf{b}_o \in \mathbb{R}^{V_e}$ is the output bias, and $\mathbf{m} \in \mathbb{R}^H$ is derived as

$$\mathbf{m} = \text{layer\_norm}\big(\text{gelu}(\mathbf{W}_f \mathbf{h} + \mathbf{b}_f)\big), \quad (2)$$

where $\mathbf{h} \in \mathbb{R}^H$ is the output embedding corresponding to the masked entity, $\mathbf{W}_f \in \mathbb{R}^{H \times H}$ is the weight matrix, $\mathbf{b}_f \in \mathbb{R}^H$ is the bias, gelu($\cdot$) is the gelu activation function (Hendrycks and Gimpel, 2016), and layer_norm($\cdot$) is the layer normalization function (Lei Ba et al., 2016).

3.3 Training

We used the same transformer architecture adopted in the BERT$_{\text{LARGE}}$ model (Devlin et al., 2019). We initialized the parameters of our model that were common with BERT (i.e., parameters in the transformer encoder and the embeddings for words) using the uncased version of the pretrained BERT$_{\text{LARGE}}$ model.[1] Other parameters, namely, the parameters in the MEP and the embeddings for entities, were initialized randomly.

The model was trained via iterations over Wikipedia pages in a random order for seven epochs. We treated the hyperlinks as entity annotations, and masked 30% of all entities at random. The input text was tokenized to words using the BERT's tokenizer with its vocabulary consisting of $V_w = 30,000$ words. Similar to Ganea and Hofmann (2017), we built an entity vocabulary consisting of $V_e = 128,040$ entities, which were contained in the entity candidates in the datasets used in our experiments. We optimized the model by maximizing the log likelihood of MEP's predictions using Adam (Kingma and Ba, 2014). Further details are provided in Appendix A.

---

[1] We initialized $\mathbf{C}_{word}$ using BERT's segment embedding for sentence A.

Algorithm 1: Algorithm of our global ED model.
___
Input: Words and mentions $m_1, \ldots m_N$ in the input
   document
Initialize: $e_i \leftarrow$ [MASK]$, i = 1 \ldots N$
repeat N **times**
    For all mentions, obtain entity predictions
    $\hat{e}_1 \ldots \hat{e}_N$ using Eq.(2) and Eq.(3) using words
    and entities $e_1, \ldots, e_N$ as inputs
    Select a mention m_j that has the most confident
    prediction in all unresolved mentions
    $e_j \leftarrow \hat{e}_j$
end
return $\{e_1, \ldots, e_N\}$
___

4 Our ED Model

We describe our ED model in this section.

4.1 Local ED Model

Given an input document with N mentions and their K entity candidates, our local ED model first creates an input sequence consisting of words in the document, and N masked entity tokens corresponding to the mentions in the document. Then, the model computes the embedding $\mathbf{m}' \in \mathbb{R}^H$ for each mention using Eq. (2), and predicts the entity for each mention using the softmax function over its K entity candidates:

$$\hat{\mathbf{y}}_{ED} = \text{softmax}(\mathbf{B}^* \mathbf{m}' + \mathbf{b}_o^*), \qquad (3)$$

where $\mathbf{B}^* \in \mathbb{R}^{K \times H}$ and $\mathbf{b}_o^* \in \mathbb{R}^K$ consist of the entity token embeddings and the output bias values corresponding to the entity candidates, respectively. Note that $\mathbf{B}^*$ and $\mathbf{b}_o^*$ are the subsets of $\mathbf{B}$ and $\mathbf{b}_o$, respectively. This model is denoted as **local** in the remainder of the paper.

4.2 Global ED Model

Our global model addresses ED by resolving mentions sequentially for N steps. The model is described in Algorithm 1. First, the model initializes the entity of each mention using the [MASK] token. Then, for each step, the model predicts an entity for each mention, selects a mention with the highest probability produced by the softmax function in Eq.(3) in all unresolved mentions, and resolves the selected mention by assigning the predicted entity to the mention. This model is denoted as **confidence-order** in the remainder of the paper. Furthermore, we test a baseline model that selects a mention by its order of appearance in the document and denote it by **natural-order**.

Name	Train	Accuracy
Yamada et al. (2016)	✓	91.5
Ganea and Hofmann (2017)	✓	92.22±0.14
Yang et al. (2018)	✓	93.0
Le and Titov (2018)	✓	93.07±0.27
Cao et al. (2018)		80
Fang et al. (2019)	✓	94.3
Shahbazi et al. (2019)	✓	93.46±0.14
Le and Titov (2019)		89.66±0.16
Broscheit (2019)	✓	87.9
Yang et al. (2019) (DCA-SL)	✓	94.64±0.2
Yang et al. (2019) (DCA-RL)	✓	93.73±0.2
Our (confidence-order)	✓	**95.04±0.24**
Our (natural-order)	✓	94.76±0.26
Our (local)	✓	94.49±0.22
Our (confidence-order)		92.42
Our (natural-order)		91.68
Our (local)		90.80

Table 1: In-KB accuracy on the CoNLL dataset. The 95% confidence intervals over five runs are also reported if available. **Train:** whether the model is trained using the training set of the CoNLL dataset.

5 Experiments

We test the proposed ED models using six standard ED datasets: AIDA-CoNLL[2] (CoNLL) (Hoffart et al., 2011), MSNBC (MSB), AQUAINT (AQ), ACE2004 (ACE), WNED-CWEB (CW), and WNED-WIKI (WW) (Guo and Barbosa, 2018). We consider only the mentions that refer to valid entities in Wikipedia. For all datasets, we use the *KB+YAGO* entity candidates and their associated $\hat{p}(e|m)$ (Ganea and Hofmann, 2017), and use the top 30 candidates based on $\hat{p}(e|m)$. We split a document if it is longer than 512 words, which is the maximum word length of the BERT model. We report the in-KB accuracy for the CoNLL dataset, and the micro F1 score (averaged per mention) for the other datasets.

Furthermore, we optionally fine-tune the model by maximizing the log likelihood of the ED predictions ($\hat{\mathbf{y}}_{ED}$) using the training set of the CoNLL dataset. We mask 90% of the mentions and fix the entity token embeddings ($\mathbf{B}$ and $\mathbf{B}^*$) and the output bias ($\mathbf{b}_o$ and $\mathbf{b}_o^*$). The model is trained for two epochs using Adam. Additional details are provided in Appendix B.

5.1 Results and Analysis

Table 1 presents the results of the CoNLL dataset. Our global models successfully outperformed all the recent strong models, including models based on ELMo (Shahbazi et al., 2019) and BERT (Broscheit, 2019). Furthermore, our confidence-

___
[2]We used the *test_b* set of the CoNLL dataset.

Name	Train	MSB	AQ	ACE	CW	WW	Avg.
Ganea and Hofmann (2017)	✓	93.7	88.5	88.5	77.9	77.5	85.2
Yang et al. (2018)	✓	92.6	89.9	88.5	**81.8**	79.2	86.4
Le and Titov (2018)	✓	93.9	88.3	89.9	77.5	78.0	85.5
Cao et al. (2018)		-	87	88	-	86	-
Fang et al. (2019)	✓	92.8	87.5	91.2	78.5	82.8	86.6
Shahbazi et al. (2019)	✓	92.3	90.1	88.7	78.4	79.8	85.9
Le and Titov (2019)		92.2	90.7	88.1	78.2	81.7	86.2
Yang et al. (2019) (DCA-SL)	✓	94.6	87.4	89.4	73.5	78.2	84.6
Yang et al. (2019) (DCA-RL)	✓	93.8	88.3	90.1	75.6	78.8	85.3
Our (confidence-order)		**96.3**	**93.5**	**91.9**	78.9	89.1	**89.9**
Our (natural-order)		96.1	92.9	**91.9**	78.4	**89.2**	89.6
Our (local)		96.1	91.9	**91.9**	78.4	88.8	89.3
Our (confidence-order)	✓	94.1	91.5	90.7	78.3	87.6	88.4
Our (natural-order)	✓	94.1	90.9	90.7	78.3	87.4	88.3
Our (local)	✓	93.9	90.8	90.7	78.2	87.2	88.2

Table 2: Micro F1 scores on the five ED datasets. **Train:** whether the model is trained using the training set of the CoNLL dataset.

# annotations	confidence-order	natural-order	local	G&H2017
0	1.0	1.0	1.0	0.8
1–10	95.55	95.55	95.55	91.93
11–50	96.98	96.70	96.43	92.44
≥51	96.64	96.38	95.80	94.21

Table 3: In-KB accuracy on the CoNLL dataset split by the frequency in Wikipedia entity annotations. Our models were fine-tuned using the CoNLL dataset. **G&H2017**: The results of Ganea and Hofmann (2017).

order model trained only on our Wikipedia-based annotations outperformed two recent models trained on the in-domain training set of the CoNLL dataset, namely, Yamada et al. (2016) and Ganea and Hofmann (2017).

Table 2 presents the results of the datasets other than the CoNLL dataset. Our models trained only on our Wikipedia-based annotations outperformed recent strong models on the MSB, AQ, ACE, and WW datasets. Additionally, we also tested the performance of our models fine-tuned on the CoNLL dataset, and found that fine-tuning generally degraded the performance on these five datasets.

Furthermore, our local model performed equally or worse in comparison with our global models on all datasets. This clearly demonstrates the effectiveness of using global contextual information even if the local contextual information is modeled based on expressive contextualized embeddings. Moreover, the natural-order model performed worse than the confidence-order model on most datasets.

Additionally, our models performed relatively worse on the CW dataset. We consider that our model failed to capture important contextual information because this dataset is significantly longer on average than other datasets, i.e., approximately 1,700 words per document on average, which is more than thrice longer than the maximum word length of our model (i.e., 512 words). We also consider that Yang et al. (2018) achieved excellent performance on this specific dataset because their model is based on various hand-engineered features capturing document-level contextual information.

To investigate how the global contextual information helped our model to improve performance, we manually analyzed the difference between the predictions of the local, natural-order, and confidence-order models. The CoNLL dataset was used to fine-tune and test the models.

The local model often failed to resolve mentions of common names referring to specific entities (e.g., "New York" referring to the basketball team *New York Knicks*). Global models were generally better to resolve such mentions because of the presence of strong global contextual information (e.g., mentions referring to basketball teams).

Furthermore, we found that the CoNLL dataset contains mentions that require a highly detailed context to resolve. For example, a mention "Matthew Burke" can refer to two different former Australian rugby players. Although the local and natural-order models incorrectly resolved this mention to the player who has the larger number of occurrences in our Wikipedia-based annotations, the confidence-order model successfully resolved this mention by disambiguating its contextual mentions, including his colleague players, in advance. We provide detailed inference of the corresponding document in Appendix C.

Next, we examined if our model learned effective embeddings for rare entities using the CoNLL dataset. Following Ganea and Hofmann (2017), we used the mentions of which entity candidates contain their gold entities, and measured the performance by dividing the mentions based on the frequency of their entities in the Wikipedia annotations used to train the embeddings. As presented in Table 3, our models achieved enhanced performance to predict rare entities.

6 Conclusions

We proposed a global ED model based on contextualized embeddings trained using Wikipedia. Our experimental results demonstrate the effectiveness of our model across a wide range of ED datasets.

References

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.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural Collective Entity Linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge Text and Knowledge by Learning Multi-Prototype Entity Mention Embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1623–1633.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint Entity Linking with Deep Reinforcement Learning. In *The World Wide Web Conference*, WWW '19, pages 438–447.

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.

Zhaochen Guo and Denilson Barbosa. 2018. Robust Named Entity Disambiguation with Random Walks. *Semantic Web*, 9(4):459–479.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415v3*.

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 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980v9*.

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.

Phong Le and Ivan Titov. 2019. Boosting Entity Linking Performance by Leveraging Unlabeled Documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1935–1945.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450v1*.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 2013 International Conference on Learning Representations*, pages 1–12.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. 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 2227–2237.

Hamed Shahbazi, Xiaoli Z. Fern, Reza Ghaeini, Rasha Obeidat, and Prasad Tadepalli. 2019. Entity-aware ELMo: Learning Contextual Entity Representation for Entity Disambiguation. *arXiv preprint arXiv:1908.05762v2*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

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.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning Distributed Representations of Texts and Entities from Knowledge Base. *Transactions of the Association for Computational Linguistics*, 5:397–411.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning Dynamic Context Augmentation for Global Entity Linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 271–281.

Yi Yang, Ozan Irsoy, and Kazi Shefaet Rahman. 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)*, pages 777–786.

A Details of Training of Contextualized Embeddings

As the input corpus for training our contextualized embeddings, we used the December 2018 version of Wikipedia, comprising approximately 3.5 billion words and 11 million entity annotations. We generated input sequences by splitting the content of each page into sequences comprising ≤ 512 words and their entity annotations (i.e., hyperlinks).

To stabilize the training, we updated only those parameters that were randomly initialized (i.e., fixed the parameters initialized using BERT) at the first epoch, and updated all the parameters in the remaining six epochs. We implemented the model using PyTorch, and the training took approximately ten days using eight Tesla V100 GPUs.

The hyper-parameters used in the training are detailed in Table 4.

Name	Value
number of hidden layers	24
hidden size	1024
attention heads	16
attention head size	64
activation function	gelu
maximum word length	512
batch size	2048
learning rate (1st epoch)	5e-4
learning rate decay (1st epoch)	none
warmup steps (1st epoch)	1000
learning rate	5e-5
learning rate decay	linear
warmup steps	1000
dropout	0.1
weight decay	0.01
gradient clipping	1.0
adam β_1	0.9
adam β_2	0.999
adam ϵ	1e-6

Table 4: Hyper-parameters used for training our contextualized embeddings.

B Details of Fine-tuning on CoNLL Dataset

The hyper-parameters used in the fine-tuning on the CoNLL dataset are detailed in Table 5. We selected these hyper-parameters from the search space described in Devlin et al. (2019) based on the accuracy on the development set of the CoNLL dataset.

Name	Value
maximum word length	512
number of epochs	2
batch size	16
learning rate	2e-5
learning rate decay	linear
warmup proportion	0.1
dropout	0.1
weight decay	0.01
gradient clipping	1.0
adam β_1	0.9
adam β_2	0.999
adam ϵ	1e-6

Table 5: Hyper-parameters during fine-tuning on the CoNLL dataset.

C Example of Inference by Confidence-order Model

Figure 2 shows an example of the inference performed by our confidence-order model fine-tuned on the CoNLL dataset. The document was obtained from the test set of the CoNLL dataset. As shown in the figure, the model started with unambiguous player names to recognize the topic of the document, and subsequently resolved the mentions that were challenging to resolve.

Notably, the model correctly resolved the mention "Nigel Walker" to the corresponding former rugby player instead of a football player, and the mention "Matthew Burke" to the correct former Australian rugby player born in 1973 instead of the former Australian rugby player born in 1964, by resolving other contextual mentions, including their colleague players in advance. These two mentions are denoted in red in the figure. Note that our local model failed to resolve both mentions, and our natural-order model failed to resolve "Mattew Burke."

Document:

"Campo has a massive following in this country and has had the public with him ever since he first played here in 1984," said Andrew, also likely to be making his final **20:** Twickenham appearance. On
tour, **17:** Australia have won all four tests against **46:** Italy, **47:** Scotland, **48:** Ireland and **45:** Wales, and scored 414 points at an average of almost 35 points a game. League duties restricted the **28:** Barbarians' selectorial options but they still boast 13 internationals including **44:** England full-back **16:** Tim Stimpson and recalled wing **22:** Tony Underwood, plus **12:** All Black forwards **25:** Ian Jones and **14:** Norm Hewitt.
Teams: **27:** Barbarians - 15 - **7:** Tim Stimpson (**31:** England); 14 - **50:** Nigel Walker (**36:** Wales), 13 - **1:** Allan Bateman (**32:** Wales), 12 - **10:** Gregor Townsend (**39:** Scotland), 11 - **4:** Tony Underwood (**34:** England); 10 - **17:** Rob Andrew (**33:** England), 9 - **2:** Rob Howley (**35:** Wales); 8 - **15:** Scott Quinnell (**37:** Wales), 7 - **8:** Neil Back (**38:** England), 6 - **19:** Dale McIntosh (**41:** Pontypridd), 5 - **24:** Ian Jones (**51:** New Zealand), 4 - **11:** Craig Quinnell (**40:** Wales), 3 - **5:** Darren Garforth (**42:** Leicester), 2 - **18:** Norm Hewitt (**52:** New Zealand), 1 - **3:** Nick Popplewell (**49:** Ireland). **43:** Australia - 15 - **53:** Matthew Burke; 14 - **9:** Joe Roff, 13 - **26:** Daniel Herbert, 12 - **20:** Tim Horan (captain), 11 - **23:** David Campese; 10 - **29:** Pat Howard, 9 - Sam Payne; 8 - Michael Brial, 7 - **30:** David Wilson, 6 - **13:** Owen Finegan, 5 - **21:** David Giffin, 4 - Tim Gavin, 3 - Andrew Blades, 2 - Marco Caputo, 1 - **6:** Dan Crowley.

Order of Inference by Confidence-order Model:

Allan Bateman ➜ Rob Howley ➜ Nick Popplewell ➜ Tony Underwood ➜ Darren Garforth ➜ Dan Crowley ➜ Tim Stimpson ➜ Neil Back ➜ Joe Roff ➜ Gregor Townsend ➜ Craig Quinnell ➜ All Black ➜ Owen Finegan ➜ Norm Hewitt ➜ Scott Quinnell ➜ Tim Stimpson ➜ Australia ➜ Norm Hewitt ➜ Dale McIntosh ➜ Tim Horan ➜ David Giffin ➜ Tony Underwood ➜ David Campese ➜ Ian Jones ➜ Ian Jones ➜ Daniel Herbert ➜ Barbarians ➜ Barbarians ➜ Pat Howard ➜ David Wilson ➜ England ➜ Wales ➜ England ➜ England ➜ Wales ➜ Wales ➜ Wales ➜ England ➜ Scotland ➜ Wales ➜ Pontypridd ➜ Leicester ➜ Australia ➜ England ➜ Wales ➜ Italy ➜ Scotland ➜ Ireland ➜ Ireland ➜ Nigel Walker ➜ New Zealand ➜ New Zealand ➜ Matthew Burke

Figure 2: An illustrative example showing the inference performed by our fine-tuned confidence-order model on a document in the CoNLL dataset. Mentions are shown as underlined. Numbers in bold face represent the selection order of the confidence-order model.