Natural Language Engineering (2024), 1–22
doi:10.1017/S1351324924000019

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Anisotropic span embeddings and the negative impact of higher-order inference for coreference resolution: An empirical analysis

Feng Hou[1], Ruili Wang[1], See-Kiong Ng[2], Fangyi Zhu[2], Michael Witbrock[3], Steven F. Cahan[4], Lily Chen[5] and Xiaoyun Jia[6]

[1]School of Natural and Computational Sciences, Massey University, Auckland, New Zealand, [2]Institute of Data Science, National University of Singapore, Singapore, [3]School of Computer Science, University of Auckland, Auckland, New Zealand, [4]University of Auckland Business School, Auckland, New Zealand, [5]Research School of Accounting, College of Business & Economics, Australian National University, Australia, and [6]Institute of Governance, Shandong University, Qingdao, China

Corresponding author: Ruili Wang; Email: ruili.wang@massey.ac.nz

(Received 21 August 2021; revised 5 January 2024; accepted 5 January 2024)

Abstract

Coreference resolution is the task of identifying and clustering mentions that refer to the same entity in a document. Based on state-of-the-art deep learning approaches, end-to-end coreference resolution considers all spans as candidate mentions and tackles mention detection and coreference resolution simultaneously. Recently, researchers have attempted to incorporate document-level context using higher-order inference (HOI) to improve end-to-end coreference resolution. However, HOI methods have been shown to have marginal or even negative impact on coreference resolution. In this paper, we reveal the reasons for the negative impact of HOI coreference resolution. Contextualized representations (e.g., those produced by BERT) for building span embeddings have been shown to be highly anisotropic. We show that HOI actually increases and thus worsens the anisotropy of span embeddings and makes it difficult to distinguish between related but distinct entities (e.g., *pilots* and *flight attendants*). Instead of using HOI, we propose two methods, Less-Anisotropic Internal Representations (LAIR) and Data Augmentation with Document Synthesis and Mention Swap (DSMS), to learn less-anisotropic span embeddings for coreference resolution. LAIR uses a linear aggregation of the first layer and the topmost layer of contextualized embeddings. DSMS generates more diversified examples of related but distinct entities by synthesizing documents and by mention swapping. Our experiments show that less-anisotropic span embeddings improve the performance significantly (+2.8 F1 gain on the OntoNotes benchmark) reaching new state-of-the-art performance on the GAP dataset.

Keywords: Coreference resolution; higher-order inference; anisotropic span embeddings; contextualized representations

1. Introduction

Coreference resolution (Sukthanker *et al.* 2020) is the task of identifying and clustering mentions in a document that refer to the same entity. Coreference resolution has important applications in areas such as question answering and relation extraction. Traditional coreference resolution models (Popescu-Belis, 2003; Raghunathan *et al.* 2010; Durrett and Klein, 2013; Wiseman, Rush, and Shieber, 2016; Clark and Manning, 2016a, 2016b) work in a pipelined fashion. They usually process the task in two stages: mention detection and coreference resolution. Mention detection identifies the entity mentions in a document; coreference resolution clusters mentions that refer

to the same entity. At both stages, syntactic parsers are relied on to build complicated hand-engineered features. Consequently, traditional pipelines suffer from cascading errors and are difficult to generalize to new datasets and languages (Lee *et al.* 2017b).

To overcome the shortcomings of pipeline models, Lee *et al.* (2017b) proposed the first end-to-end model that tackles mention detection and coreference resolution simultaneously. They consider all spans as mention candidates and use two scoring functions to learn which spans are entity mentions and which are their coreferential antecedents. The training objective is to optimize the marginal log-likelihood of all correct antecedents implied by the gold-standard clustering. To control model complexity, they use a unary mention scoring function to prune the space of spans and antecedents and a pairwise antecedent scoring function to compute the softmax distribution over antecedents for each span. Both scoring functions are simple feed-forward neural networks, and the inputs to both scoring functions are the learned span embeddings. Thus, the core of end-to-end neural coreference resolution models is the learning of span embeddings. A bi-directional LSTM was used to generate the embedded representations of spans (Lee *et al.* 2017b; Zhang *et al.* 2018).

The aforementioned end-to-end coreference resolution model (Lee *et al.* 2017b) is a "first-order" model that only considers local context and does not directly incorporate any information about the entities to which the spans might refer. Thus, first-order models may suffer from consistency errors. Higher-order inference (HOI) methods were therefore proposed to incorporate some document-level information. Most such HOI methods, such as Attended Antecedent (AA) (Lee, He, and Zettlemoyer, 2018; Joshi *et al.* 2019, 2020), Entity Equalization (Kantor and Globerson, 2019), and Span Clustering (Xu and Choi, 2020), are based on span refinement. These methods iteratively refine span embeddings using global context from the other spans. Following the success of contextualized representations, ELMo (Peters *et al.* 2018; Lee *et al.*, 2018), BERT (Devlin *et al.* 2019; Joshi *et al.* 2019; Kantor and Globerson, 2019), and SpanBERT (Joshi *et al.* 2020) have been used to build span embeddings for higher-order models.

This research is motivated by the following three recent findings: (i) It was shown that HOI methods have marginal or even negative impact on coreference resolution (Xu and Choi, 2020), but the reason is not clear. (ii) Contextualized representations have recently been shown to be anisotropic (i.e., not directionally uniform in the vector space) (Ethayarajh, 2019), especially the representations in the topmost layer. (iii) It has been found that isotropy is beneficial and that less-anisotropic embeddings could lead to large improvements on downstream NLP tasks (Mu, Bhat, and Viswanath, 2018; Ethayarajh, 2019).

In this paper, we reveal the reasons for the negative impact of HOI on coreference resolution. We show that HOI actually increases and thus worsens the anisotropy (i.e., lack of directional uniformity) of span embeddings and makes it difficult to distinguish between related but distinct entities (e.g., *pilots* and *flight attendants*). We propose two methods, Less-Anisotropic Internal Representations (LAIR) and Data Augmentation with Document Synthesis and Mention Swap (DSMS), to learn less-anisotropic span embeddings. LAIR uses the linear aggregation of the first layer and the topmost layer of contextualized embeddings. DSMS generates more diversified examples of related but distinct entities by synthesizing documents and mention swap. We conduct comprehensive experiments to understand the reasons behind HOI's negative impact on end-to-end coreference resolution using various state-of-the-art contextualized representations based span embeddings:

1. The discrepancy of HOI impact on ELMo (Lee *et al.*, 2018; Peters *et al.*, 2018), BERT (Devlin *et al.*, 2019; Joshi *et al.*, 2019; Kantor and Globerson, 2019), SpanBERT (Joshi *et al.*, 2020), and ELECTRA (Clark *et al.* 2020) can be explained by their degree of anisotropy and contextualization. The span embeddings from SpanBERT and ELECTRA are more anisotropic and are encoded with longer context than ELMo and BERT.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

2. The span embeddings from SpanBERT and ELECTRA without higher-order refinement are thus less anisotropic and more effective than high-order span embeddings.
3. Furthermore, the less-anisotropic span embeddings from SpanBERT and ELECTRA can significantly improve performance. Using LAIR and DSMS, SpanBERT and ELECTRA gained + 0.7 F1 and + 2.8F1 on the OntoNotes benchmark respectively.
4. LAIR directly builds less-anisotropic span embeddings and achieves significant improvements over HOI-based methods, while DSMS alone achieves only marginal improvements without the incorporation of LAIR.

2. Background of end-to-end neural coreference resolution

2.1. Task formulation

End-to-end coreference resolution is the state-of-the-art approach based on deep learning. It tackles mention detection and coreference resolution simultaneously by span ranking; thus, it is formulated as a task of assigning antecedents a_i for each span i. A possible span candidate is any continuous N-gram within a sentence. The set of possible assignments a_i is $\mathscr{A}_i = \{1, \ldots, i-1, \varepsilon\}$, where ε is a "dummy" antecedent. If span i is assigned to a non-dummy antecedent, span j, then we have $a_i = j$. If span i is assigned to a dummy antecedent ε, then it indicates one of two scenarios: (1) span i is not an entity mention; (2) span i is the first mention of a new entity (cluster). Through transitivity of coreferent antecedents, these assignment decisions induce clusters of entities over the document.

2.2. First-order coreference resolution

The first-order end-to-end coreference resolution model (Lee *et al.* 2017b) independently ranks each pair of spans using a pairwise scoring function $s(i, j)$. These scores are used to compute the antecedent distribution $P(a_i)$ for each span i:

$$P(a_i) = \frac{e^{s(i,a_i)}}{\sum_{j \in \mathscr{A}_i} e^{s(i,j)}} \tag{1}$$

The coreferent score $s(i, j)$ for a pair of spans includes three factors: (1) $s_m(i)$, the score of span i for being a mention, (2) $s_m(j)$, the score of span j for being a mention, (3) $s_a(i, j)$, the score of span j for being an antecedent of i:

$$s(i,j) = \begin{cases} 0 & j = \varepsilon \\ s_m(i) + s_m(j) + s_a(i,j) & j \neq \varepsilon \end{cases} \tag{2}$$

The scoring functions s_m and s_a take span representations $\mathbf{g}$ as input:

$$\begin{aligned} s_m(i) &= \mathbf{w}_m \cdot FFNN_m(\mathbf{g}_i) \\ s_a(i,j) &= \mathbf{w}_a \cdot FFNN_a\left(\left[\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i,j)\right]\right) \end{aligned} \tag{3}$$

where $\cdot$ denotes the dot product; $\circ$ denotes element-wise multiplication; *FFNN* denotes a feed-forward neural network; and $\phi(i, j)$ represents speaker and meta-data features (e.g., genre information and the distance between the two spans).

However, it is intractable to score every pair of spans in a document. There are $O(W^2)$ spans of potential mentions in a document (W is the number of words). Comparing every pair would be $O(W^4)$ complexity. Thus, pruning is performed according to the mention scores $s_m(i)$ to reduce the spans that are unlikely to be an entity mention.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

2.3. Higher-order coreference resolution

The first-order coreference resolution model only considers pairs of spans, and does not directly incorporate any information about the entities to which the spans might belong. Thus, first-order models may suffer from consistency errors. HOI methods were proposed to incorporate some document-level information. Span-refinement-based HOI methods, such as AA (Lee *et al.*, 2018; Joshi *et al.* 2019, 2020), Entity Equalization (Kantor and Globerson, 2019), and Span Clustering (Xu and Choi, 2020), iteratively refine span embeddings $\mathbf{g}_i$ using global context. Cluster Merging (Xu and Choi, 2020) is a score-based HOI method. It does not update span embeddings directly, but updates the coreferent scores using a latent cluster score.

AA (Lee *et al.*, 2018) was the first HOI method. It iteratively refines the span representations $\mathbf{g}_i^n$ of span i at the nth iteration, using information from antecedents. The refined span representations are used to compute the refined antecedent distribution $P_n(a_i)$:

$$P_n(a_i) = \frac{e^{s\left(\mathbf{g}_i^n, \mathbf{g}_{a_i}^n\right)}}{\sum_{j \in \mathscr{A}_i} e^{s\left(\mathbf{g}_i^n, \mathbf{g}_j^n\right)}} \tag{4}$$

At each iteration, the expected antecedent representation $\mathbf{a}_i^n$ of each span i is computed using the current antecedent distribution $P_n(a_i)$ as an attention mechanism:

$$\mathbf{a}_i^n = \sum_{j \in \mathscr{A}_i} P_n(j)\mathbf{g}_j^n \tag{5}$$

The current span representation $\mathbf{g}_i^n$ is then updated via interpolation with its expected antecedent representation $\mathbf{a}_i^n$:

$$\mathbf{g}_i^{n+1} = \mathbf{f}_i^n \circ \mathbf{g}_i^n + \left(1 - \mathbf{f}_i^n\right) \circ \mathbf{a}_i^n \tag{6}$$

where $\mathbf{f}_i^n = \sigma\left(\mathbf{W}_f\left[\mathbf{g}_i^n, \mathbf{a}_i^n\right]\right)$ is a learned gate vector, and $\circ$ denotes element-wise multiplication. Thus, the span representation $\mathbf{g}_i^{n+1}$ at iteration $n+1$ is an element-wise weighted average of the current span representation $\mathbf{g}_i^n$ and its direct antecedents.

2.4. Building span embeddings from contextualized representations

The core of end-to-end neural coreference resolution is the learning of vectorized representations of text spans. The span representation $\mathbf{g}_i$ of span i is usually the concatenation of four vectors (Lee *et al.* 2017b) as follows:

$$\mathbf{g}_i = \left[\mathbf{g}^*_{START(i)}, \mathbf{g}^*_{END(i)}, \hat{\mathbf{g}}_i, \phi(i)\right] \tag{7}$$

where $START(i)$ and $END(i)$ are the start position and end position of span i, respectively, **Boundary** representations $\mathbf{g}^*_{START(i)}, \mathbf{g}^*_{END(i)}$ are the vector representations of word $START(i)$ and $END(i)$, respectively. **Internal** representation $\hat{\mathbf{g}}_i$ is a weighted sum of word vectors in span i, and $\phi(i)$ is a feature vector encoding the size of span i.

Assuming the vector representations of each word are $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ (T is the length of the document), Lee *et al.* (2017b), Zhang *et al.* (2018) and Lee *et al.* (2018) use a bi-directional LSTM to build the first three vectors of the span representation $\mathbf{g}_i$:

$$\begin{aligned} \mathbf{g}^*_{START(i)} &= BiLSTM\left(\mathbf{x}_{START(i)}\right) \\ \mathbf{g}^*_{END(i)} &= BiLSTM\left(\mathbf{x}_{END(i)}\right) \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot \mathbf{x}_t \end{aligned} \tag{8}$$

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

where $a_{i,t}$ is a learned weight computed from $BiLSTM(\mathbf{x}_t)$, and $\mathbf{x}_t$ can be GloVe (Jeffrey Pennington, Socher, and Manning, 2014; Lee *et al.* 2017b), ELMo (Lee *et al.*, 2018; Peters *et al.*, 2018), or the concatenation of GloVe and CNN character embeddings (Santos and Zadrozny, 2014; Zhang *et al.*, 2018).

Following the success of contextualized representations, Kantor and Globerson (2019), and Joshi *et al.* (2019) replace the LSTM-based encoder with BERT (Devlin *et al.*, 2019). They either use BERT in a convolutional mode (Kantor and Globerson, 2019) or split the documents into fixed length before applying BERT (Joshi *et al.*, 2019). Kantor and Globerson (2019) use a learnable weighted average of the **last four layers** of BERT to build span representations. Joshi *et al.* (2019) use the **topmost layer** output of BERT to build span representations:

$$\begin{aligned} \mathbf{g}^*_{START(i)} &= BERT_{l=top}\left(w_{START(i)}\right) \\ \mathbf{g}^*_{END(i)} &= BERT_{l=top}\left(w_{END(i)}\right) \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot BERT_{l=top}(w_t) \end{aligned} \tag{9}$$

where $BERT_l(w_i)$ is the lth layer contextualized embeddings of token w_i, and $a_{i,t}$ is a learned weight computed from the topmost layer output. Documents are split into segments of fixed length and BERT is applied to each segment. Two variants of splitting were proposed: overlap and independent (non-overlapping). Surprisingly, independent splitting performs better.

3. Related work

3.1. Effectiveness of higher-order inference

Lee *et al.* (2018) propose the first HOI method, AA, for coreference resolution. Their experiments show that AA improved ELMo's performance on OntoNotes by + 0.4 F1. Kantor and Globerson (2019) show that removing AA improved BERT's performance slightly. Experiments by Xu and Choi (2020) show that span-refinement-based HOI methods, AA, Entity Equalization (Kantor and Globerson, 2019), and Span Clustering (Xu and Choi, 2020), have a negative impact on contextualized encoders, such as SpanBERT. However, the reasons behind this are not clear. In particular, the discrepancy of HOI impact among ELMo, BERT, and SpanBERT is not explained.

3.2. Measures of anisotropy and contextuality

Ethayarajh (2019) proposes to measure how contextual and anisotropic a word representation is using three different metrics: self-similarity, intra-sentence similarity, and random similarity. We adopt their measures to gauge how anisotropic and how contextual a word representation or a span embedding is and show that the HOI methods actually increase the anisotropy of span embeddings.

3.3. Data augmentation for coreference resolution

Wu *et al.* (2020) use existing question answering datasets for **out-of-domain** data augmentation for coreference resolution by transforming the coreference resolution task into a question answering task. To reduce anisotropy with examples of related but distinct entities, we use synthesized documents for **in-domain** data augmentation.

3.4. Embeddings aggregation

Arora *et al.* (2018) hypothesize that the global word embedding is a linear combination of its sense embeddings. They show that senses can be recovered through sparse coding. Mu *et al.* (2017) show that senses and word embeddings are linearly related and sense sub-spaces tend to intersect

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

along a line. Yaghoobzadeh *et al.* (2019) probe the aggregated word embeddings of polysemous words for semantic classes. They created a WIKI-PSE corpus, where word and semantic class pairs were annotated using Wikipedia anchor links, for example "apple" has two semantic classes: *food* and *organization*. A separate embedding for each semantic class was learned based on the WIKI-PSE corpus. They find that the linearly aggregated embeddings of polysemous words represent their semantic classes well. Our previous work (Hou *et al.* 2020) shows that linearly aggregated entity embeddings can improve the performance of entity linking. In this research, we use linear aggregations of contextualized representations from different layers of Transformer (Vaswani *et al.* 2017)-based encoders to learn less-anisotropic span embeddings.

4. Revealing the reason for the negative impact of HOI

4.1. What is anisotropy

Anisotropy means that the elements of vectorized representations are not uniformly distributed with respect to direction. Instead, they occupy a narrow cone in the vector space. Its opposite, *isotropy*, has both theoretical and empirical benefits, for example it allows for stronger "self-normalization" during training (Arora, Liang, and Ma, 2017) and improves performance on downstream tasks (Mu *et al.*, 2018).

4.2. Measure of anisotropy of span embeddings

For contextualized word representations, the degree of anisotropy is measured by the average cosine similarity between the representations of randomly sampled words from different contexts (Ethayarajh, 2019). We adopt this to measure the anisotropy, $RandomSim(D)$, of span embeddings in a document D, as follows.

Let D be a document that contains m spans of entity mentions $\{s_1, s_2, \ldots, s_m\}$. Let $\mathbf{g}_i$ be the span embedding of span s_i. The *random similarity* between these spans is

$$RandomSim(D) = \frac{1}{m^2 - m} \sum_i \sum_{j \neq i} cos\left(\mathbf{g}_i, \mathbf{g}_j\right) \tag{10}$$

where *cos* denotes the cosine similarity.

4.3. HOI generates more anisotropic span embeddings

In this subsection, we prove and show that HOI makes the generated span embeddings more anisotropic. According to Equation (10), we need to prove that the HOI refined span embeddings $\mathbf{g}_i^{n+1}, \mathbf{g}_j^{n+1}$ at iteration $n+1$ are more similar than $\mathbf{g}_i^n, \mathbf{g}_j^n$ at iteration n. Based on the definition $cos(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b} / \|\mathbf{a}\| \times \|\mathbf{b}\|$ and the supposition that $\|\mathbf{g}_i^{n+1}\| = \|\mathbf{g}_i^n\|$, we can say that $\mathbf{g}_i^{n+1}, \mathbf{g}_j^{n+1}$ are more similar than $\mathbf{g}_i^n, \mathbf{g}_j^n$ if $\mathbf{g}_i^{n+1} \cdot \mathbf{g}_j^{n+1} > \mathbf{g}_i^n \cdot \mathbf{g}_j^n$.

Combining Equations (5) and (6) and treat vector $\mathbf{f}_i^n$ as scalar f_i^n, we get

$$\begin{aligned}
\mathbf{g}_i^{n+1} &= f_i^n \mathbf{g}_i^n + \left(1 - f_i^n\right) \sum_{k=1}^{m} P_n(k) \mathbf{g}_k^n \approx f_i^n \mathbf{g}_i^n + \left(1 - f_i^n\right) \left(P_n(i) \mathbf{g}_i^n + P_n(j) \mathbf{g}_j^n\right) \\
\mathbf{g}_j^{n+1} &= f_j^n \mathbf{g}_j^n + \left(1 - f_j^n\right) \sum_{k=1}^{m} P'_n(k) \mathbf{g}_k^n \approx f_j^n \mathbf{g}_j^n + \left(1 - f_j^n\right) \left(P'_n(i) \mathbf{g}_i^n + P'_n(j) \mathbf{g}_j^n\right)
\end{aligned} \tag{11}$$

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

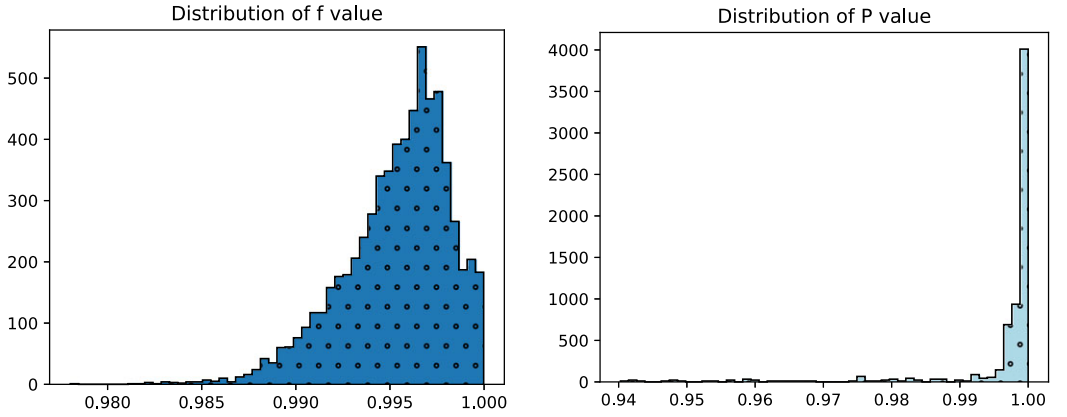

Figure 1. The histogram distributions of f and P values. We recorded 6564 f and P values respectively during the last two epochs of training C2F+BERT-base+**AA**. f is the average across the embedding dimension; P is the max value of antecedents.

Treating f_i^n and f_j^n equally as f, $P_n(i)$ and $P'_n(j)$ equally as P, $P_n(j)$ and $P'_n(i)$ equally as $\hat{P}$, we get

$$\mathbf{g}_i^{n+1} \cdot \mathbf{g}_j^{n+1} \approx \left(f\mathbf{g}_i^n + (1-f)\left(P\mathbf{g}_i^n + \hat{P}\mathbf{g}_j^n\right)\right) \cdot \left(f\mathbf{g}_j^n + (1-f)\left(\hat{P}\mathbf{g}_i^n + P\mathbf{g}_j^n\right)\right) \tag{12}$$

and

$$\begin{aligned}\mathbf{g}_i^{n+1} \cdot \mathbf{g}_j^{n+1} \approx &\left(f^2 + 2f(1-f)P + (1-f)^2P^2 + (1-f)^2\hat{P}^2\right)\mathbf{g}_i^n \cdot \mathbf{g}_j^n \\ &+ 2f(1-f)\hat{P} + 2(1-f)^2P\hat{P}\end{aligned} \tag{13}$$

Figure 1 shows the histogram distribution of the f and P values. Practically, as shown in Figure 1, $f, P \to 1$, and as $P, \hat{P}$ are the elements of a softmax list, $P + \hat{P} \approx 1$, $\hat{P} \to 0$. Thus, in Equation (13), the former part approximately equals $\mathbf{g}_i^n \cdot \mathbf{g}_j^n$, and the latter part is not zero. Thus, $\mathbf{g}_i^{n+1} \cdot \mathbf{g}_j^{n+1} > \mathbf{g}_i^n \cdot \mathbf{g}_j^n$. This proves that the HOI methods tend to generate more anisotropic span embeddings. From Equation (13), we can see that the more iterations we refine the span embeddings for, the more anisotropic the span embeddings will become.

To concretely show the degree of anisotropy of span embeddings generated by different HOI methods, we compute and record the random similarity (Equation (10)) at each epoch during training. Figure 2 displays the recorded average cosine similarity of each epoch. As shown in Figure 2, the span embeddings generated by HOI methods are always more anisotropic than the span embeddings generated without HOI.

According to Equations (5) and (6), higher-order span refinement is essentially injecting information from other spans into a span's embedding. Thus, unrelated spans may become more similar (and hence, more *anisotropic*).

4.4. Why anisotropic span embeddings are undesirable

Span embeddings encode the contextual information (and some meta-information about spans) of the textual spans that are potentially an entity mention. Spans in the same sentence or segment will have similar span embeddings. Thus, some degree of anisotropy is also an indicator of contextuality (Ethayarajh, 2019). To explain this, we compute the measures of contextualization and anisotropy proposed by Ethayarajh (2019) and visualize these measures in Figures 3, 4, and 5. Figure 3 shows the random cosine similarities (the degree of anisotropy) of each layer.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

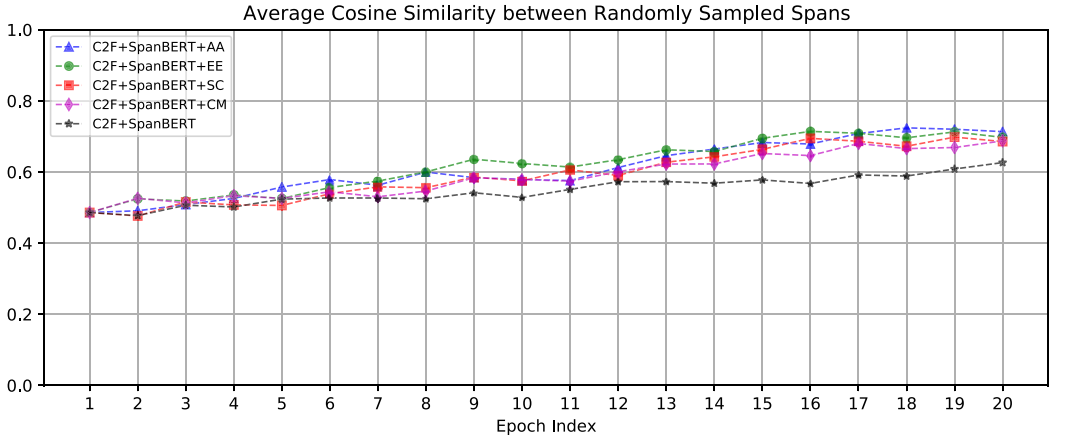

Figure 2. The degree of anisotropy of span embeddings is measured by the average cosine similarity between uniformly randomly sampled spans. (Figures 2-6 are generated using the method of Ethayarajh (2019)).

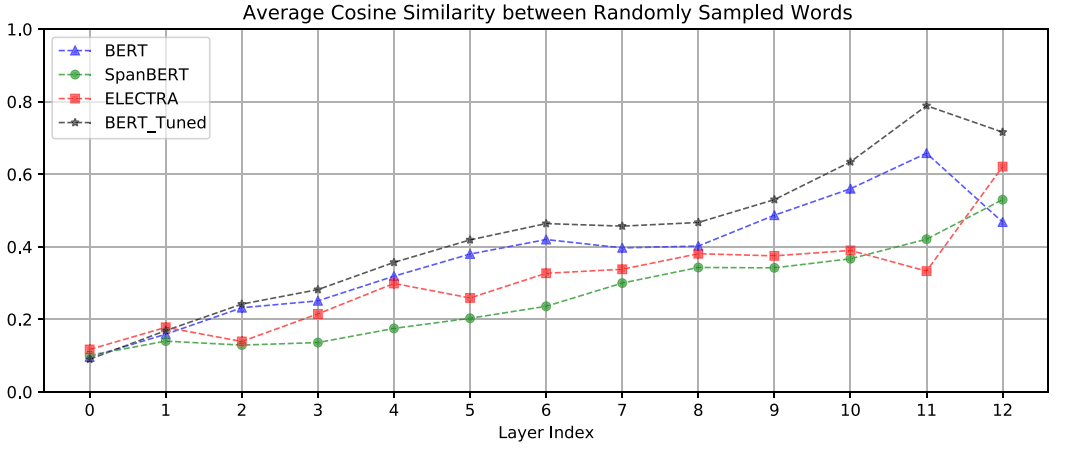

Figure 3. For contextualized word representations, the degree of anisotropy is measured by the average cosine similarity between uniformly randomly sampled words. The higher the layer, the more anisotropic. Embeddings of layer 0 are the input layer word embeddings.

Figures 4 and 5 show the intra-sentence similarities and self-similarities of each layer, respectively. Both the intra-sentence similarity and the self-similarity are contextualization measures proposed by Ethayarajh (2019). As shown in Figures 4 and 5, the higher layer representations are more context-specific, that is the top layer representations are more contextualized. As shown in Figure 3, the top layer representations are also more anisotropic. However, if the embeddings of the spans in the same document are too anisotropic, that is the span embeddings occupy a narrow cone in the vector space, related but distinct entities (e.g., *pilots* and *flight attendants*) will become difficult to distinguish.

This can be explained by the example of *pilots* and *flight attendants* by Lee *et al.* (2017b). As Lee *et al.* (2017b) analyzed, related but distinct entities appear in similar contexts and likely have nearby word (span) embeddings. Thus, their neural coreference resolution model predicts a false positive link between *pilots* and *flight attendants*. Related entities are grouped in a local cluster in the vector space, while unrelated entities in different contexts are grouped in different clusters in the vector space. Thus, unrelated entities are more distinguishable than the related entities. However, extremely isotropic (not anisotropic) embeddings are also undesirable. For a

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

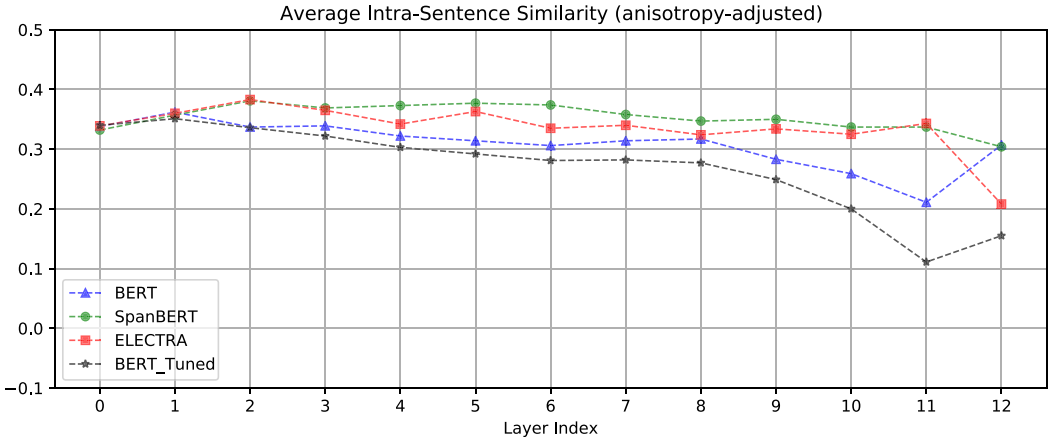

Figure 4. Intra-sentence similarities of contextualized representations. The intra-sentence similarity is the average cosine similarity between each word representation in a sentence and their mean.

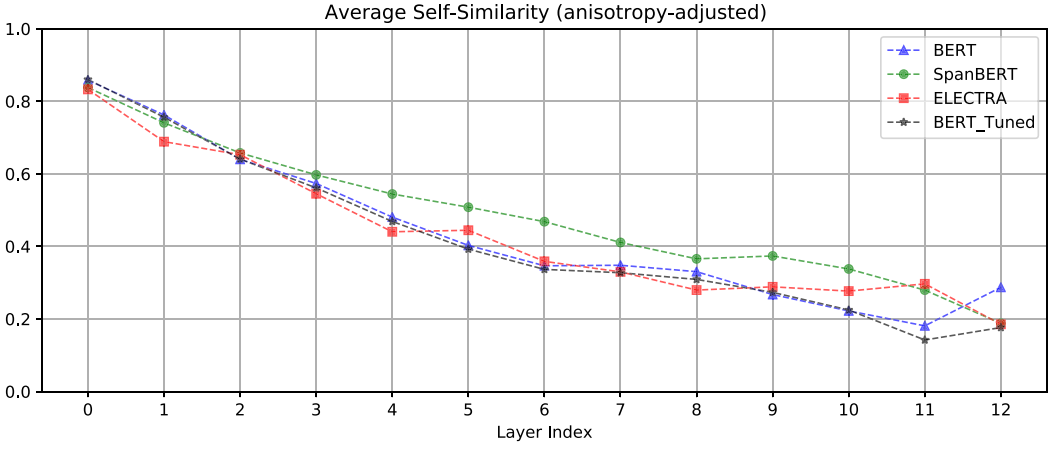

Figure 5. Self-similarities of contextualized representations. Self-similarity is the average cosine similarity between representations of the same word in different contexts.

vocabulary with V words, the extremely isotropic embeddings would be the embeddings in a $|V|$ dimension with each word occupying a single dimension. Such extremely isotropic embeddings cannot capture the similarity between words and lose contextualization. As more contextualized representations tend to be anisotropic (Ethayarajh, 2019), we need to achieve a trade-off between isotropy and contextualization.

In summary, we believe that the negative impact of HOI is caused by worsening the anisotropy of span embeddings without incorporating more contexts. We propose methods that learn less-anisotropic span embeddings in Section 5 and perform comprehensive experimental analysis in Section 6.

5. Learning less-aniostropic span embeddings

5.1. Less-anisotropic internal representations (LAIR)

Contextualized word representations are more context-specific in the higher layers. However, Ethayarajh (2019) shows that these contextualized word representations are anisotropic, especially in the higher layers. Building span embeddings directly from the output-layer contextualized word

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

representations will cause anisotropy. To reduce the degree of anisotropy while retaining contextual information, we use a linear aggregation of the contextualized embeddings of the first layer and the topmost layer. As the boundary representations need to encode more contextual information, we only use aggregated embeddings for the internal representation $\hat{\mathbf{g}}_i$:

$$\begin{aligned} AT(w_t) &= \alpha \circ T_{l=1}(w_t) + (1-\alpha) \circ T_{l=top}(w_t) \\ \mathrm{a}_t &= \mathbf{w}_\mathrm{a} \cdot FFNN_\mathrm{a}(AT(w_t)) \\ a_{i,t} &= \frac{exp(\mathrm{a}_t)}{\sum_{k=START(i)}^{END(i)} exp(\mathrm{a}_k)} \\ \hat{\mathbf{g}}_i &= \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot AT(w_t) \end{aligned} \tag{14}$$

where $T_l(w)$ is the lth layer Transformer-based (Vaswani *et al.*, 2017) contextualized embedding of token w, AT is the aggregated representation, and the scalar α is the weight of the first layer embedding.

Aggregating the first layer and the topmost layer embeddings can reduce anisotropy without substantial loss of contextual information, because the topmost layer encodes the most contextual information, and the first layer is the least anisotropic (Ethayarajh, 2019).

5.2. Data augmentation with document synthesis and mention swap (DSMS)

As we mentioned in Section 4.4, anisotropic span embeddings make it difficult to distinguish related but distinct entities. We propose to learn less-anisotropic span embeddings by using a more diversified set of examples of related but distinct entities, generated by synthesizing documents and by mention swapping. Algorithm 1 describes the complete process for this data augmentation method, the details of which are as follows.

5.2.1. synthesizing documents

We split each training document into segments of 512 word pieces (Wu *et al.* 2016) and then combine segments from two different documents to synthesize two-segment-long training "documents." The process is as follows:

1) For each pair of documents (d_1, d_2) of the same genre, compute their TF-IDF cosine similarity s. Sort the list of (d_1, d_2, s) from the largest to the smallest according to similarity s.
2) Sequentially select a pair of documents (d_1, d_2) from the ordered list $[(d_1, d_2, s)]$. If both documents have not been selected, then build a synthesized document of two segments using the second segment of d_1 and the first segment of d_2. Mentions and clusters are updated to sew both segments together. These steps are detailed below.
3) Repeat step 2) until no documents can be generated.

The two documents used to synthesize a new document possibly contain coreferent mentions. Such documents can be viewed as noisy examples. Such noisy examples help reduce the similarity of span embeddings (e.g., reduce the similarity of the two span embeddings of "Biden") and improve the generalization of resolution models.

Why sort the list of (d_1, d_2, s) *by similarity?.* By synthesizing documents using segments from similar real documents, we can create training examples with distinct (non-coreferent) entities of

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Algorithm 1: Data augmentation for learning less-anisotropic span embeddings

Input: Segmented Original Documents D^o of genre g
Output: Segmented Synthesized Documents D^s of genre g
for *each pair of documents* (d_1, d_2) *in* D^o **do**
  Compute the TF-IDF cosine similarity s between d_1 and d_2;
  Append (d_1, d_2, s) to list S;
end
Sort elements of S from largest s to smallest s;
Initialize empty list $\overline{D^o}$;
for *each element* (d_1, d_2, s) *in* S **do**
  if d_1 *in* $\overline{D^o}$ *or* d_2 *in* $\overline{D^o}$ **then**
    continue;
  else
    Append d_1 and d_2 to $\overline{D^o}$;
    Synthesize a new document d = [*secondSegment* (d_1), *firstSegment* (d_2)];
    Update the Mentions and clusters in d;
    $d = MentionSwap(d)$;
    Append d to D^s;
  end
end
return D^s;

similar contexts. Our hypothesis is that we can learn less-anisotropic span embeddings from such training examples.

Selecting documents for synthesis. We select documents of the same genre to synthesize documents; a document is used only once for synthesizing documents. Since a synthesized document consists of two segments from different documents, N documents of a genre will generate $N/2$ synthesized documents.

Updating mentions and clusters. A synthesized document consists of two segments from different documents, which are sewn together by simply updating the start and end positions of the mentions. Mentions that are not in the selected segments are removed from the clusters.

5.2.2. Mention swap

For a synthesized document, we randomly select a cluster with more than one mention and then randomly select two different mentions from that cluster and swap them. We hypothesize that more swaps might introduce noisy information. Thus, we do only one swap in a synthesized document. Mention swapping is performed as follows:

1) Randomly select 2 mentions in the same cluster;
2) Swap the token IDs of these mentions;
3) Update the start and end positions of both mentions.

5.2.3. Learning less-anisotropic span embeddings

Data augmentation can help learn less-anisotropic span embeddings in the following two ways:

- The synthesized documents consist of two independent (no coreferences) segments with similar contexts. The pairwise score function (Equation (3)) computes the coreference

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

scores of all pairs of spans in the documents. Thus, the synthesized documents provide examples of non-coreferential mentions (spans) in similar contexts. Such examples help learn less-anisotropic span embedding for spans in similar contexts (anisotropic embeddings are too similar to distinguish).

- Mention swap changes the contexts of a pair of mentions and thus can provide different combinations of boundary representations and internal representations (Equation (7)) of a span.

6. Experiments

6.1. Data sets and evaluation metrics

6.1.1. Document level coreference resolution: ontoNotes

OntoNotes (English) is a document-level dataset from the CoNLL-2012 shared task (Pradhan *et al.* 2012) on coreference resolution. It consists of 2,802/343/348 train/development/test documents of different genres, such as newswire, magazine articles, broadcast news, broadcast conversations, etc.

Evaluation metrics. The main evaluation metric is the average F1 of three metrics (Recasens and Hovy, 2011): MUC (Vilain *et al.* 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_{\phi 4}$ (Luo, 2005) on the test set according to the official CoNLL-2012 evaluation scripts.

6.1.2. Paragraph level coreference resolution: GAP

GAP (Webster *et al.* 2018) is a human-labeled corpus of ambiguous pronoun-name pairs derived from Wikipedia snippets. Examples in the GAP dataset fit within a single segment, thus obviating the need for cross-segment inference.

Evaluation metrics. The metrics are the F1 score on **M**asculine and **F**eminine examples, **O**verall, and the **B**ias factor (i.e., F/M). Following Webster *et al.* (2018) and Joshi *et al.* (2019), the coreference resolution system is trained on OntoNotes and only the testing is performed on GAP. The dataset and scoring script are available on GitHub.[a]

6.1.3. Data sets for gauging contextualized representations

Like Ethayarajh (2019), we use the data from the SemEval Semantic Textual Similarity tasks from the years 2012–2016 (Agirre *et al.* 2012, 2013, 2014, 2015). The other settings remain the same.

6.2. Implementation and hyperparameters

Before performing the experiments on coreference resolution, we gauged the anisotropy and contextuality of the contextualized representations from BERT, SpanBERT, and ELECTRA using the code[b] from Ethayarajh (2019) on the data sets mentioned in Section 6.1.3. The results for anisotropy are shown in Figure 3. The measures of contextuality are intra-sentence similarity and self-similarity, shown in Figures 4 and 5 respectively.

We split the OntoNotes English documents into segments of 512 word pieces (Wu *et al.*, 2016), performing data augmentation only on the training set. To compare the effects of our methods on multiple encoders, we also carried out experiments on ELECTRA with uncased vocabulary, using the HuggingFace PyTorch version of ELECTRA[c] (discriminator).

[a] https://github.com/google-research-datasets/gap-coreference
[b] https://github.com/kawine/contextual
[c] https://github.com/huggingface/transformers

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Table 1. Hyperparameter settings for our experiments

Hyperparameter description	Setting value
# Training segments	3
Segments length	512 (word pieces)
Top span ration	0.4
Max span width	30 (word pieces)
Equation (3) hidden size	3000
Use span width prior	True

Our code is based on the implementation in Joshi *et al.* (2019) and Xu and Choi (2020). We use similar hyperparameters, except that we introduce a new hyperparameter: the weight of the first layer embeddings, α, in Equation (14). We chose α as follows: For SpanBERT, we tried α over the range from 0.1 to 0.6 with a 0.1 increment. Since ELECTRA is more anisotropic than SpanBERT (as shown in Figure 3), we tried larger values of α for ELECTRA. All the α values are chosen based on the development set of OntoNotes. These hyperparameters for our experiments are listed in Table 1.

6.3. Baselines

To validate our methods for reducing anisotropy, we compared our methods with systems that combine coarse-to-fine antecedent pruning (C2F) (Lee *et al.*, 2018) with the following HOI modules: **AA** (Lee *et al.*, 2018), Entity Equalization (Kantor and Globerson, 2019), Span Clustering (Xu and Choi, 2020), and cluster merging (**CM**) (Xu and Choi, 2020).

We also applied LAIR + DSMS to C2F+SpanBERT+**CM**, but found this approach did not improve performance and therefore removed the **CM** module (i.e., LAIR + DSMS + C2F+SpanBERT). This not only could improve performance, but could also significantly accelerate training.

6.4. Results

The results on OntoNotes are listed in Table 2. For SpanBERT, (LAIR + DSMS) offers an improvement of + 0.7 F1 over the system that does not use any HOI; the improvement over the system that uses AA is + 1.0 F1. For ELECTRA, (LAIR + DSMS) boosts the performance by + 2.8 F1 over the system that does not use any HOI. The results also show that DSMS alone achieves only marginal improvements without the incorporation of LAIR. We analyze the reasons for this phenomenon in Section 6.5.1.

The results on GAP are listed in Table 3. Combined with (LAIR + DSMS), ELECTRA achieves new state-of-the-art for performance on the GAP dataset.

6.5. Analysis and findings

6.5.1. Effects of LAIR and DSMS

The effects of LAIR and DSMS for learning less-anisotropic span embeddings can be assessed by measuring the degree of anisotropy in those learned span embeddings and relating this to performance. Figure 6 shows the average cosine similarity between randomly sampled spans based on the span embeddings generated by SpanBERT with different settings. The similarities are the

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Table 2. Results on the test set of the OntoNotes English data from the CoNLL-2012 shared task. HOI modules are in bold. The rightmost column is the main evaluation metric, the average F1 of MUC, B^3, $CEAF_{\phi_4}$. BERT, SpanBERT, and ELECTRA are the large model. The results of the models with citations are copied verbatim from the original papers

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
E2E (Lee *et al.* 2017b)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
C2F + ELMo (Lee *et al.*, 2018)	80.4	79.0	80.1	71.0	70.0	70.5	67.5	67.2	67.3	72.6
C2F + ELMo + **AA** (Lee *et al.*, 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
C2F + BERT (Kantor and Globerson, 2019)	82.6	83.5	83.0	73.6	75.4	74.5	71.6	71.6	71.6	76.4
C2F + BERT + **EE** (Kantor and Globerson, 2019)	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
C2F + BERT + **AA** (Joshi *et al.*, 2019)	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
C2F + SpanBERT + **AA** (Joshi *et al.*, 2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
C2F + SpanBERT + **SC** (Xu and Choi, 2020)	85.5	85.2	85.4	78.4	78.5	78.4	76.5	74.1	75.2	79.7
C2F + SpanBERT + **CM** (Xu and Choi, 2020)[1]	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
C2F + SpanBERT (Xu and Choi, 2020)	85.7	85.3	85.5	78.6	78.6	78.6	76.8	74.8	75.8	79.9
+ DSMS	85.9	85.5	85.7	78.8	78.5	78.7	76.6	74.4	75.5	79.9
+ (LAIR, $\alpha = 0.2$)	86.2	85.0	85.6	79.8	78.2	79.0	76.9	75.9	76.4	80.3
+ DSMS	**86.6**	84.4	85.5	**80.9**	77.9	79.4	77.4	76.5	76.9	80.6
C2F + ELECTRA	84.3	85.2	84.7	76.5	78.4	77.4	75.3	73.1	74.2	78.8
+ DSMS	85.3	84.8	85.1	77.9	77.1	77.6	74.8	73.7	74.2	79.0
+ (LAIR, $\alpha = 0.4$)	85.7	86.5	86.1	78.2	80.6	79.4	77.5	75.6	76.5	80.7
+ DSMS	86.1	**86.5**	**86.3**	80.1	**81.3**	**80.7**	**77.6**	**78.2**	**77.9**	**81.6**

[1]**CM** is not a span-refinement-based HOI method. The results here are copied verbatim from the original paper. However, we cannot reproduce the reported results (Avg. F1 80.2). When we train "C2F+SpanBERT+CM," we found the training is extremely slow and can achieve only marginal even negative improvements (Avg. F1 80.0 over 79.9). The marginal improvement of CM is not worth its extreme complexity.

measure of anisotropy defined in Equation (10). The less similar the spans are, the less anisotropic are the span embeddings. We can see that LAIR and DSMS effectively reduce the anisotropy of span embeddings. As shown in Table 2 and Figure 6, LAIR + DSMS achieved the best performance with the least anisotropic span embeddings.

Case study. We perform a case study on the cosine similarity between *The pilots'* and *The flight attendants* to verify our method further. This is an example in Lee *et al.* (2017b) from the OntoNotes dev set (document number: bn/cnn/02/cnn_0210_0). As shown in Figure 7, our LAIR method can effectively make the span embeddings of *The pilots'* and *The flight attendants* less similar, that is less anisotropic.

We find that applying DSMS alone achieves only marginal improvement for both SpanBERT and ELECTRA. As shown in Table 2 (11th and 15th rows) and Table 3 (8th and 11th rows), applying solely DSMS can only achieve marginal improvement. Figure 6 shows that DSMS cannot effectively reduce the degree of anisotropy of span embeddings. DSMS is essentially a data augmentation method; training with the augmented examples can only fine-tune span embeddings yet cannot significantly reduce anisotropy (as shown in Figure 6). In contrast, LAIR builds less-anisotropic span embeddings directly by incorporating the embeddings of the bottom layer, that is

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Table 3. Performance on the test set of GAP corpus. The metrics are F1 scores on **M**asculine and **F**eminine examples, **O**verall F1 score, and a **B**ias factor(F/M). BERT, SpanBERT, and ELECTRA are the large model. The models are trained using OntoNotes and tested on the test set of GAP. ∗ denotes the model is re-implemented following Joshi *et al.* (2020)

Model	M	F	B	O
E2E (Lee *et al.* 2017b)	67.2	62.2	0.92	64.7
C2F + **AA** + ELMo (Lee *et al.*, 2018)	75.8	71.1	0.94	73.5
+ BERT (Joshi *et al.*, 2019)	86.9	83.0	0.95	85.0
CorefQA + SpanBERT (Wu *et al.*, 2020)	88.9	86.1	**0.97**	87.5
C2F + SpanBERT + **AA** ∗	89.7	84.8	0.95	87.2
C2F + ELECTRA + **AA** ∗	89.1	85.2	0.96	87.2
C2F + SpanBERT	89.9	84.7	0.94	87.3
+ DSMS	89.7	85.2	0.95	87.3
+ (LAIR, DSMS)	**90.0**	85.4	0.95	87.7
C2F + ELECTRA	89.0	85.6	0.96	87.3
+ DSMS	89.6	85.1	0.95	87.3
+ (LAIR, DSMS)	89.6	**86.1**	0.96	**87.9**

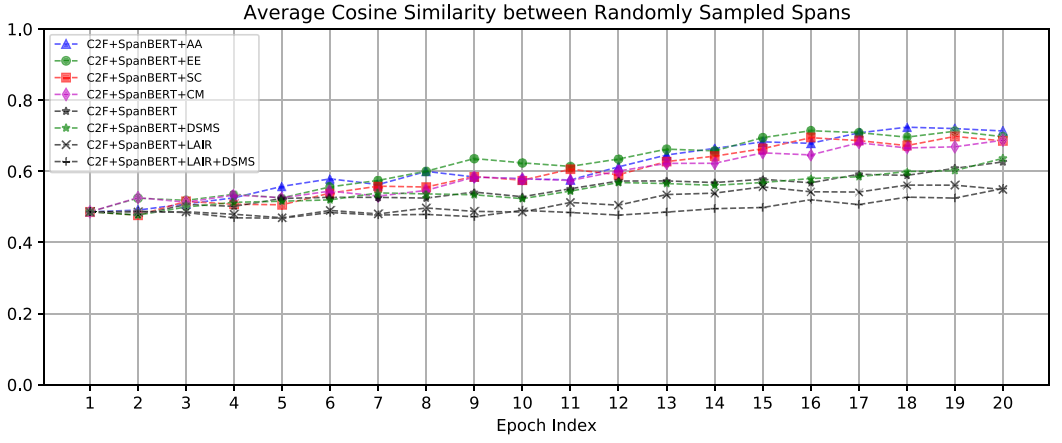

Figure 6. The average cosine similarity between randomly sampled spans. The less similar between spans, the less anisotropic are the span embeddings. LAIR and DSMS help learn less-anisotropic span embeddings.

the least anisotropic layer. Combining DSMS with LAIR can further fine-tune the less-anisotropic span embeddings and improve performances.

We also experimented with less-anisotropic **boundary** representations, but achieved slightly worse results. This is in line with the fact that the boundary representations were designed to encode a span's contextual information, while the **internal** representation was designed to encode the internal information of a span. Replacing first layer embeddings with input layer word embeddings also did not lead to improvement, indicating that the internal representation $\hat{\mathbf{g}}_i$ still needs contextual information to better encode a span's internal structure.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

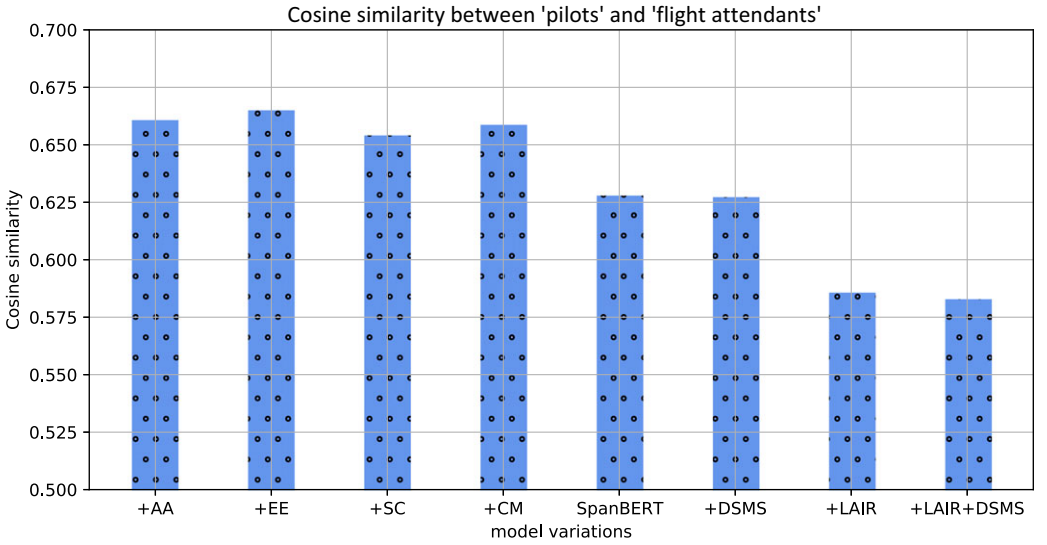

Figure 7. The cosine similarity between *The pilots'* and *The flight attendants*, an example from OntoNotes dev set.

6.5.2. LAIR and DSMS are ineffective for BERT

We could not achieve any improvement for BERT by using LAIR or DSMS. The 1st-3rd rows of Table 4 show that decreasing the HOI iterations causes the performance of BERT-base drop significantly. The 3rd-6th rows of Table 4 show that LAIR has negative impact for BERT. BERT has the following two characteristics:

- The contextualized embeddings from BERT are highly anisotropic. The embeddings of every layer are more anisotropic than those of SpanBERT and ELECTRA, as shown in Figure 3. BERT fine-tuned for coreference resolution becomes even more anisotropic.
- BERT is not capable of encoding longer contexts effectively (Joshi *et al.*, 2019). Figures 4 and 5 also corroborate that BERT-base does not encode sufficient contextual information.

Although LAIR can directly reduce the anisotropy of BERT word embeddings, applying LAIR also depletes the contextual information needed for coreference resolution. Thus, LAIR and DSMS are not effective for BERT.

6.5.3. Degree of anisotropy and α

Figure 3 shows that the bottom layer is the least anisotropic. This means increasing α (i.e., increasing the proportion of the bottom layer) will make the span embeddings less anisotropic. However, Figures 4 and 5 also show that the bottom layer contains the least contextual information. Thus, increasing α will also deplete the contextual information for coreference resolution. The results in Table 5 show that settings with $\alpha \geq 0.5$ perform poorly. Increasing α worsens performance. Entirely using the bottom layer ($\alpha = 1.0$) achieves the worst results.

As we mentioned in Section 4.4, we need to choose the best α to achieve a trade-off between isotropy and contextualization. According to Figure 3, ELECTRA is more anisotropic than SpanBERT. In Table 2, the 11th and 14th rows show that LAIR performs the best when α is set to 0.2 for SpanBERT and 0.4 for ELECTRA, respectively. We can say that the more anisotropic the embeddings are, larger the α (Equation 14) is needed for LAIR.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Table 4. Ablation studies of higher-order inference for BERT-base, ELECTRA-base, and SpanBERT-base. The metric is the average F1 score on the OntoNotes dev set and test set using different combinations of hyperparameters. The HOI method used is the Attended Antecedent Lee *et al.* (2018)

	Segment Length Segment Num	Higher-order Iterations	α Value	F1 dev	F1 test
BERT-base	128 × 10	2	0	74.4	73.9
		1	0	74.2	73.7
		0	0	72.1	71.8
		0	0.1	71.9	71.7
		0	0.2	71.8	71.7
		0	0.3	71.5	71.4
	256 × 5	2	0	73.9	73.4
		1	0	73.8	73.3
SpanBERT-base	384 × 2	2	0	77.4	77.1
		2	0.1	77.7	77.5
		1	0.1	77.9	77.6
	384 × 3	2	0	77.7	77.4
		1	0	77.8	77.5
		1	0.1	77.7	77.5
		1	0.2	77.8	77.9
	512 × 2	1	0	77.7	77.3
		1	0.2	77.4	77.3
ELECTRA-base	384 × 2	2	0	76.2	76.1
		2	0.2	77.2	77.2
		1	0	76.7	76.9
		1	0.2	77.5	77.5
	384 × 3	1	0	76.9	76.8
		1	0.1	77.3	77.1
		1	0.2	77.5	77.6
		1	0.3	77.8	77.7
	512 × 2	1	0	77.1	77.0
		1	0.1	77.7	77.4
		1	0.2	78.0	77.6
		1	0.3	78.0	77.9

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Table 5. Ablation studies on different α values for different models

	Segment Length Segment Num	Higher-order Iterations	α Value	F1 dev	F1 test
BERT-base	128 × 11	2	0	74.5	74.1
			0.1	74.2	73.9
			0.2	74.1	73.9
			0.3	73.9	73.7
			0.4	73.6	73.5
			0.5	72.4	72.2
			0.6	72.1	72.1
			0.7	71.7	71.5
			0.8	71.1	71.1
			0.9	70.7	70.6
			1.0	69.8	69.9
SpanBERT-base	384 × 3	2	0	77.7	77.4
			0.1	77.8	77.7
			0.2	77.9	77.9
			0.3	77.7	77.6
			0.4	77.5	77.3
			0.5	77.4	77.3
			0.6	77.2	77.1
			0.7	76.9	76.8
			0.8	76.7	76.6
			0.9	75.5	75.4
			1.0	74.1	74.1
ELECTRA-base	512 × 2	2	0	77.3	77.1
			0.1	77.7	77.5
			0.2	78.1	77.9
			0.3	78.1	78.0
			0.4	77.9	77.9
			0.5	77.7	77.5
			0.6	76.9	76.7
			0.7	76.2	76.2
			0.8	75.8	75.7
			0.9	74.6	74.5
			1.0	73.9	73.8

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

6.5.4. Discrepancy of higher-order inference effectiveness for ELMo, BERT, SpanBERT, and ELECTRA

We perform ablation studies to test the effectiveness of HOI and LAIR. Table 4 shows the trend of performance with the increase of α and the number of HOI iterations. Considering the tiny change of α and HOI iterations, the marginal but consistent change of performance is meaningful. When a larger α is applied, the change of F1 is manifest. HOI methods are computationally expensive. Reducing the number of HOI iterations can reduce computational complexity. A marginal improvement of F1 with a smaller number of HOI iterations is worthwhile.

As shown in Table 4, SpanBERT-base (the 10th-13th rows) and ELECTRA-base (the 18th and 20th rows) achieve better results when using fewer iterations of HOI. While BERT-base (the 1st and 2nd rows) still needs a deeper higher-order refinement of span embeddings to get the best result using segments of 128 word pieces (Wu *et al.*, 2016). Lee *et al.* (2018) showed that combining ELMo with HOI achieved + 0.4 F1 improvement. There is a discrepancy between the HOI effectiveness for ELMo, BERT, SpanBERT, and ELECTRA.

Why is HOI effective for ELMo and BERT?. For ELMo, we find ELMo is much less anisotropic and less contextualized than BERT, SpanBERT, and ELECTRA. HOI does not increase the anisotropy of ELMo-based span embeddings to a threshold where the negative effect can be seen; at the same time, it does incorporate global information.

For BERT, BERT-base achieves the best performance when using segments of 128 word pieces, as shown in Table 4 (the 1st and 7th rows). This shows BERT is not capable of encoding longer contexts (Joshi *et al.*, 2019). Figures 4 and 5 also show that BERT-base generates less contextualized word (span) embeddings than SpanBERT-base and ELECTRA-base. HOI helps BERT incorporate document-level contextual information from longer contexts, without greatly increasing anisotropy.

Why is HOI negative for spanBERT and ELECTRA?. SpanBERT and ELECTRA achieve the best results using segments of 512 word pieces. Thus, they are capable of encoding longer contextual information. Their measures of contextuality (Ethayarajh, 2019) in Figures 4 and 5 also show they are not only highly contextualized but also highly anisotropic. Applying HOI to SpanBERT or ELECTRA does not incorporate much global information but it worsens the anisotropy of the span embeddings. As Table 4 (the 12th-13th rows and the 18th-20th rows) shows, the more iterations we refine span embeddings for, the worse performance we achieve for SpanBERT and ELECTRA. This supports our analysis in Section 4.3.

6.5.5. Document segment length and contextualization

The results in Table 4 show that SpanBERT-base and ELECTRA-base can use longer document segments (384 tokens or 512 tokens) for coreference resolution, while BERT-base can only use short document segments (only 128 tokens). Figures 4 and 5 demonstrate that SpanBERT-base and ELECTRA-base can generate more contextualized representations than BERT-base. This means SpanBERT and ELECTRA are more effective for encoding longer contexts. Thus, our methods can reduce anisotropy without depleting contextualization for SpanBERT and ELECTRA.

6.5.6. Word embeddings vs Layer 1 embeddings

Figure 3 shows that the layer 0 embeddings (the input layer word embeddings) and the layer 1 embeddings are the least anisotropic; we therefore also experimented with the input layer word embeddings, but without improvement. The difference is that the layer 1 embeddings encode contextual information, showing that the internal representation $\hat{\mathbf{g}}_i$ still needs contextual information to better encode a span's internal structure.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

7. Conclusions

In this paper, we revealed the reasons for the negative impact of HOI on coreference resolution. We showed that HOI actually increases and thus worsens the anisotropy of span embeddings and makes it difficult to distinguish between related but distinct entities (e.g., *pilots* and *flight attendants*). We proposed two methods, LAIR and DSMS, to learn less-anisotropic span embeddings for coreference resolution. LAIR uses the linear aggregation of the first layer and the topmost layer of the Transformer-based contextualized embeddings, such as SpanBERT and ELECTRA. DSMS generates more diversified examples of related but distinct entities by synthesizing documents and by mention swapping.

We evaluated LAIR and DSMS on the OntoNotes benchmark for document-level coreference resolution and the GAP data sets for paragraph-level coreference resolution. Our experiments showed that less-anisotropic span embeddings improve the performance significantly (+2.8 F1 gain on OntoNotes benchmark), achieving new state-of-the-art for performance on the GAP dataset.

We performed ablation studies to test the effectiveness of HOI and LAIR, revealing the reason for the negative impact of HOI methods. We found that (i) HOI-negative encoders such as ELECTRA and SpanBERT output longer-context-encoded contextualized but anisotropic representations, and LAIR can reduce anisotropy without depleting contextual information; (ii) HOI-positive encoders such as ELMo and BERT only encode shorter contexts, and HOI helps incorporate global contextual information from longer contexts.

Acknowledgments. This work was supported by the 2020 Catalyst: Strategic NZ-Singapore Data Science Research Programme Fund, MBIE, New Zealand.

Competing interests. The authors declare none.

References

Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Lopez-Gazpio I., Maritxalar M., Mihalcea R., Rigau G., Uria L. and Wiebe J. (2015). *SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability*. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado: Association for Computational Linguistics, pp. 252–263.

Agirre E., Banea C., Cardie C., Cer D., Diab M., Gonzalez-Agirre A., Guo W., Mihalcea R., Rigau G. and Wiebe J. (2014). *SemEval-2014 task 10: Multilingual semantic textual similarity*. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, Dublin, Ireland: Association for Computational Linguistics, pp. 81–91.

Agirre E., Cer D., Diab M. and Gonzalez-Agirre A. (2012). *SemEval-2012 task 6: A pilot on semantic textual similarity*, ∗SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Montréal, Canada: Association for Computational Linguistics, pp. 385–393.

Agirre E., Cer D., Diab M., Gonzalez-Agirre A. and Guo W. (2013). ∗*SEM. 2013 shared task: Semantic textual similarity*. In *Second Joint Conference on Lexical and Computational Semantics (∗SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 32–43.

Arora S., Li Y., Liang Y., Ma T. and Risteski A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics* **6**, 483–495.

Arora S., Liang Y. and Ma T. (2017). *A simple but tough-to-beat baseline for sentence embeddings*. In *Proceedings of International Conference on Learning Representations*.

Bagga A. and Baldwin B. (1998). *Algorithms for scoring coreference chains*. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, Granada, vol. **1**, pp. 563–566.

Clark K., Luong M.-T., Le Q. V. and Manning C. D. (2020). *ELECTRA: Pre-training text encoders as discriminators rather than generators*. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Clark K. and Manning C. D. (2016a). *Deep reinforcement learning for mention-ranking coreference models*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, pp. 2256–2262.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Clark K. and Manning C. D. (2016b). *Improving coreference resolution by learning entity-level distributed representations*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, pp. 643–653.

Devlin J., **Chang M.-W.**, **Lee K. and Toutanova K.** (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 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, vol. **1**, pp. 4171–4186.

Durrett G. and Klein D. (2013). *Easy victories and uphill battles in coreference resolution*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, pp. 1971–1982.

Ethayarajh K. (2019). *How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings*. 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)*, Hong Kong, China: Association for Computational Linguistics, pp. 55–65.

Hou F., **Wang R.**, **He J. and Zhou Y.** (2020). *Improving entity linking through semantic reinforced entity embeddings*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 6843–6848.

Joshi M., **Chen D.**, **Liu Y.**, **Weld D. S.**, **Zettlemoyer L. and Levy O.** (2020). Spanbert: improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics* **8**, 64–77.

Joshi M., **Levy O.**, **Zettlemoyer L. and Weld D.** (2019). *BERT for coreference resolution: Baselines and analysis*. 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)*, Hong Kong, China: Association for Computational Linguistics, pp. 5803–5808.

Kantor B. and Globerson A. (2019). *Coreference resolution with entity equalization*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, pp. 673–677.

Lee K., **He L.**, **Lewis M. and Zettlemoyer L.** (2017b). *End-to-end neural coreference resolution*. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, pp. 188–197.

Lee K., **He L. and Zettlemoyer L.** (2018). *Higher-order coreference resolution with coarse-to-fine inference*, (Short Papers), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Short Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, vol. **2**, pp. 687–692.

Lee H., **Surdeanu M. and Jurafsky D.** (2017a). A scaffolding approach to coreference resolution integrating statistical and rule-based models. *Natural Language Engineering* **23**(5), 733–762.

Luo X. (2005). *On coreference resolution performance metrics*. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 25–32.

Mu J., **Bhat S. and Viswanath P.** (2017). *Geometry of polysemy*. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Mu J., **Bhat S. and Viswanath P.** (2018). *All-but-the-top: Simple and effective postprocessing for word representations*. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Pennington J., **Socher R. and Manning C. D.** (2014). GloVe: global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Peters M., **Neumann M.**, **Iyyer M.**, **Gardner M.**, **Clark C.**, **Lee K. and Zettlemoyer L.** (2018). *Deep contextualized word representations*, (Long Papers), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, vol **1**, pp. 2227–2237.

Popescu-Belis A. (2003). Evaluation-driven design of a robust coreference resolution system. *Natural Language Engineering* **9**(3), 281–306.

Pradhan S., **Moschitti A.**, **Xue N.**, **Uryupina O. and Zhang Y.** (2012). *Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes*. In *Joint Conference on EMNLP and CoNLL-Shared Task*, Association for Computational Linguistics, pp. 1–40.

Raghunathan K., **Lee H.**, **Rangarajan S.**, **Chambers N.**, **Surdeanu M.**, **Jurafsky D. and Manning C.** (2010). *A multi-pass sieve for coreference resolution*. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA: Association for Computational Linguistics, pp. 492–501.

Recasens M. and Hovy E. (2011). Blanc: implementing the rand index for coreference evaluation. *Natural Language Engineering* **17**(4), 485–510.

https://doi.org/10.1017/S1351324924000019 Published online by Cambridge University Press

Santos C. D. and Zadrozny B. (2014). *Learning character-level representations for part-of-speech tagging*. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1818–1826.

Sukthanker R., **Poria S.**, **Cambria E. and Thirunavukarasu R.** (2020). Anaphora and coreference resolution: a review. *Information Fusion* **59**, 139–162.

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*, pp. 5998–6008.

Vilain M., **Burger J.**, **Aberdeen J.**, **Connolly D. and Hirschman L.** (1995). *A model-theoretic coreference scoring scheme*. In *Proceedings of the 6th Conference on Message Understanding*, Association for Computational Linguistics, pp. 45–52.

Webster K., **Recasens M.**, **Axelrod V. and Baldridge J.** (2018). Mind the GAP: a balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics* **6**, 605–617.

Wiseman S., **Rush A. M. and Shieber S. M.** (2016). *Learning global features for coreference resolution*. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, pp. 994–1004.

Wu Y., **Schuster M.**, **Chen Z.**, **Le Q. V.**, **Norouzi M.**, **Macherey W.**, **Krikun M.**, **Cao Y.**, **Gao Q.**, **Macherey K.** and et al. (2016). Google's neural machine translation system: bridging the gap between human and machine translation. cs.CL, 1–23, arXiv preprint arXiv: 1609.08144.

Wu W., **Wang F.**, **Yuan A.**, **Wu F. and Li J.** (2020). *CorefQA: Coreference resolution as query-based span prediction*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 6953–6963.

Xu L. and Choi J. D. (2020). *Revealing the myth of higher-order inference in coreference resolution*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 8527–8533.

Yaghoobzadeh Y., **Kann K.**, **Hazen T. J.**, **Agirre E. and Schütze H.** (2019). *Probing for semantic classes: Diagnosing the meaning content of word embeddings*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, pp. 5740–5753.

Zhang R., **Nogueira dos Santos C.**, **Yasunaga M.**, **Xiang B. and Radev D.** (2018). *Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, pp. 102–107.

Cite this article: Hou F., Wang R., Ng S.-K., Zhu F., Witbrock M., Cahan S. F., Chen L. and Jia X. Anisotropic span embeddings and the negative impact of higher-order inference for coreference resolution: An empirical analysis. *Natural Language Engineering* https://doi.org/10.1017/S1351324924000019