# Towards Consistent Document-level Entity Linking: Joint Models for Entity Linking and Coreference Resolution

**Klim Zaporojets, Johannes Deleu, Yiwei Jiang, Thomas Demeester, Chris Develder**
Ghent University – imec, IDLab
Ghent, Belgium
{first_name.last_name}@ugent.be

**Abstract**

We consider the task of document-level entity linking (EL), where it is important to make consistent decisions for entity mentions over the full document jointly. We aim to leverage explicit "connections" among mentions within the document itself: we propose to join EL and coreference resolution (coref) in a *single* structured prediction task over directed trees and use a globally normalized model to solve it. This contrasts with related works where two separate models are trained for each of the tasks and additional logic is required to merge the outputs. Experimental results on two datasets show a boost of up to +5% F1-score on both coref and EL tasks, compared to their standalone counterparts. For a subset of hard cases, with individual mentions lacking the correct EL in their candidate entity list, we obtain a +50% increase in accuracy.[1]

## 1 Introduction

In this paper we explore a principled approach to solve entity linking (EL) jointly with coreference resolution (coref). Concretely, we formulate coref+EL as a *single* structured task over directed trees that conceives EL and coref as two complementary components: a coreferenced cluster can only be linked to a single entity or NIL (i.e., a non-linkable entity), and all mentions linking to the same entity are coreferent. This contrasts with previous attempts to join coref+EL (Hajishirzi et al., 2013; Dutta and Weikum, 2015; Angell et al., 2021) where coref and EL models are trained separately and additional logic is required to merge the predictions of both tasks.

Our first approach (Local in Fig. 1(a)) is motivated by current state-of-the-art coreference resolution models (Joshi et al., 2019; Wu et al., 2020) that predict a single antecedent for each span to resolve.

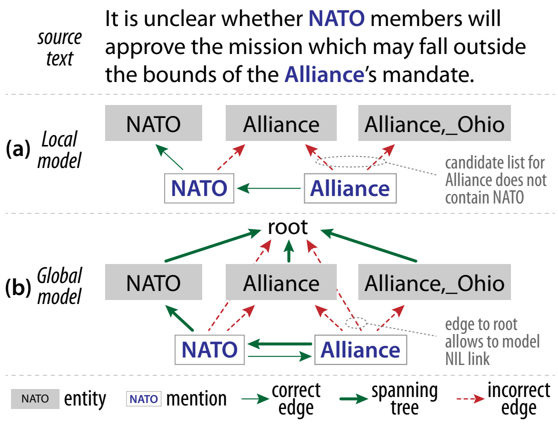


Figure 1: Illustration of our 2 explored graph models: (a) Local where edges are only allowed from spans to antecedents or candidate entities, and (b) Global where the prediction involves a spanning tree over all nodes.

We extend this architecture by also considering entity links as potential antecendents: in the example of Fig. 1, the mention "Alliance" can be either connected to its antecedent mention "NATO" or to any of its candidate links (*Alliance* or *Alliance,_Ohio*). While straightforward, this approach cannot solve cases where the first coreferenced mention does not include the correct entity in its candidate list (e.g., if the order of "NATO" and "Alliance" mentions in Fig. 1 would be reversed). We therefor propose a second approach, Global, which by construction overcomes this inherent limitation by using bidirectional connections between mentions. Because that implies cycles could be formed, we resort to solving a maximum spanning tree problem. Mentions that refer to the same entity form a cluster, represented as a subtree rooted by the single entity they link to. To encode the overall document's clusters in a single spanning tree, we introduce a virtual *root* node (see Fig. 1(b)).[2]

This paper contributes: (i) 2 architectures (Local and Global) for joint entity linking (EL) and

[1] Our code, models and AIDA$^{+}$ dataset will be released on https://github.com/klimzaporojets/consistent-EL

[2] Coreference clusters without a linked entity, i.e., a NIL cluster, have a link of a mention directly to the root.



*Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*
*Volume 2: Short Papers*, pages 778 - 784

corefence resolution, (ii) an extended AIDA dataset (Hoffart et al., 2011), adding new annotations of linked and NIL coreference clusters, (iii) experimental analysis on 2 datasets where our joint coref+EL models achieve up to +5% F1-score on both tasks compared to standalone models. We also show up to +50% in accuracy for hard cases of EL where entity mentions lack the correct entity in their candidate list.

## 2 Architecture

Our model takes as input (i) the full document text, and (ii) an *alias table* with entity candidates for each of the possible spans. Our end-to-end approach allows to jointly predict the mentions, entity links and coreference relations between them.

### 2.1 Span and Entity Representations

We use SpanBERT (base) from Joshi et al. (2020) to obtain *span* representations $\mathbf{g}_i$ for a particular span $s_i$. Similarly to Luan et al. (2019); Xu and Choi (2020), we apply an additional pruning step to keep only the top-$N$ spans based on the pruning score $\Phi_\mathrm{p}$ from a feed-forward neural net (FFNN):

$$\Phi_\mathrm{p}(s_i) = \mathrm{FFNN}_P(\mathbf{g}_i). \tag{1}$$

For a candidate entity $e_j$ of span $s_i$ we will obtain representation as $\mathbf{e}_j$ (which is further detailed in §3).

### 2.2 Joint Approaches

We propose two methods for joint coreference and EL. The first, Local, is motivated by end-to-end span-based coreference resolution models (Lee et al., 2017, 2018) that optimize the marginalized probability of the correct antecedents for each given span. We extend this local marginalization to include the span's candidate entity links. Formally, the modeled probability of $y$ (text span or candidate entity) being the antecedent of span $s_i$ is:

$$P_\mathrm{cl}(y|s_i) = \frac{\exp\big(\Phi_\mathrm{cl}(s_i, y)\big)}{\sum_{y' \in \mathcal{Y}(s_i)} \exp\big(\Phi_\mathrm{cl}(s_i, y')\big)}, \tag{2}$$

where $\mathcal{Y}(s_i)$ is the set of antecedent spans unified with the candidate entities for $s_i$. For antecedent *spans* $\{s_j : j < i\}$ the score $\Phi_\mathrm{cl}$ is defined as:

$$\Phi_\mathrm{cl}(s_i, s_j) = \Phi_\mathrm{p}(s_i) + \Phi_\mathrm{p}(s_j) + \Phi_\mathrm{c}(s_i, s_j), \tag{3}$$

$$\Phi_\mathrm{c}(s_i, s_j) = \mathrm{FFNN}_C([\mathbf{g}_i; \mathbf{g}_j; \mathbf{g}_i \odot \mathbf{g}_j; \boldsymbol{\varphi}_{i,j}]), \tag{4}$$

where $\boldsymbol{\varphi}_{i,j}$ is an embedding encoding the distance[3] between spans $s_i$ and $s_j$. Similarly, for a particular candidate *entity* $e_j$, the score $\Phi_\mathrm{cl}$ is:

$$\Phi_\mathrm{cl}(s_i, e_j) = \Phi_\mathrm{p}(s_i) + \Phi_\ell(s_i, e_j), \tag{5}$$

$$\Phi_\ell(s_i, e_j) = \mathrm{FFNN}_L([\mathbf{g}_i; \mathbf{e}_j]). \tag{6}$$

An example graph of mentions and entities with edges for which aforementioned scores $\Phi_\mathrm{cl}$ would be calculated is sketched in Fig. 1(a). While simple, this approach fails to correctly solve EL when the correct entity is only present in the candidate lists of mention spans occurring later in the text (since earlier mentions have no access to it).

To solve EL in the general case, even when the first mention does not have the correct entity, we propose bidirectional connections between mentions, thus leading to a maximum spanning tree problem in our Global approach. Here we define a score for a (sub)tree $t$, noted as $\Phi_\mathrm{tr}(t)$:

$$\Phi_\mathrm{tr}(t) = \sum_{(i,j) \in t} \Phi_\mathrm{cl}(u_i, u_j), \tag{7}$$

where $u_i$ and $u_j$ are two connected nodes (i.e., *root*, candidate entities or spans) in $t$. For a ground truth cluster $c \in C$ (with $C$ being the set of all such clusters), with its set[4] of correct subtree representations $\mathcal{T}_c$, we model the cluster's likelihood with its subtree scores. We minimize the negative log-likelihood $\mathcal{L}$ of all clusters:

$$\mathcal{L} = -\log \frac{\prod_{c \in C} \sum_{t \in \mathcal{T}_c} \exp\big(\Phi_\mathrm{tr}(t)\big)}{\sum_{t \in \mathcal{T}_{all}} \exp\big(\Phi_\mathrm{tr}(t)\big)}. \tag{8}$$

Naively enumerating all possible spanning trees ($\mathcal{T}_{all}$ or $\mathcal{T}_c$) implied by this equation is infeasible, since their number is exponentially large. We use the adapted Kirchhoff's Matrix Tree Theorem (MTT; Koo et al. (2007); Tutte (1984)) to solve this: the sum of the weights of the spanning trees in a directed graph rooted in *r* is equal to the determinant of the Laplacian matrix of the graph with the row and column corresponding to *r* removed (i.e., the *minor* of the Laplacian with respect to *r*). This way, eq. (8) can be rewritten as

$$\mathcal{L} = -\log \frac{\prod_{c \in C} \det\Big(\hat{\mathbf{L}}_c\big(\boldsymbol{\Phi}_\mathrm{cl}\big)\Big)}{\det\Big(\mathbf{L}_r\big(\boldsymbol{\Phi}_\mathrm{cl}\big)\Big)}, \tag{9}$$

[3]Measured in number of spans, after pruning.

[4]For a single cluster annotation, indeed it is possible that multiple correct trees can be drawn.

where $\mathbf{\Phi}_{\text{cl}}$ is the weighted adjacency matrix of the graph, and $\mathbf{L}_r$ is the minor of the Laplacian with respect to the root node $r$. An entry in the Laplacian matrix is calculated as

$$L_{i,j} = \begin{cases} \sum_k \exp(\Phi_{\text{cl}}(u_k, u_j)) & \text{if } i = j \\ -\exp(\Phi_{\text{cl}}(u_i, u_j)) & \text{otherwise} \end{cases}, \tag{10}$$

Similarly, $\hat{\mathbf{L}}_c$ is a *modified Laplacian* matrix where the first row is replaced with the root $r$ selection scores $\Phi_{\text{cl}}(r, u_j)$. For clarity, Appendix A presents a toy example with detailed steps to calculate the loss in eq. (9).

To calculate the scores of each of the entries $\Phi_{\text{cl}}(u_i, u_j)$ to $\mathbf{\Phi}_{\text{cl}}$ matrix in eqs. (7) and (9) for Global, we use the same approach as in Local for edges between two mention spans, or between a mention and entity. For the directed edges between the root $r$ and a candidate entity $e_j$ we choose $\Phi_{\text{cl}}(r, e_j) = 0$. Since we represent NIL clusters by edges from the mention spans directly to the root, we also need scores for them: we use eq. (3) with $\Phi_{\text{p}}(r) = 0$. We use Edmonds' algorithm (Edmonds, 1967) for decoding the maximum spanning tree.

## 3 Experimental Setup

We considered two datasets to evaluate our proposed models: DWIE (Zaporojets et al., 2021) and AIDA (Hoffart et al., 2011). Since AIDA essentially does not contain coreference information, we had to extend it by (i) adding missing mention links in order to make annotations consistent on the coreference cluster level, and (ii) annotating NIL coreference clusters. We note this extended dataset as AIDA$^+$. See Table 1 for the details.

As input to our models, for DWIE we generate spans of up to 5 tokens. For each mention span $s_i$, we find candidates from a dictionary of entity surface forms used for hyperlinks in Wikipedia. We then keep the top-16 candidates based on the prior for that surface form, as per Yamada et al. (2016, §3). Each of those candidates $e_j$ is represented using a Wikipedia2Vec embedding $\mathbf{e}_j$ (Yamada et al., 2016).[5] For AIDA$^+$, we use the spans, entity candidates, and entity representations from Kolitsas et al. (2018).[6]

To assess the performance of our joint coref+EL models Local and Global, we also provide Standalone implementations for coref and EL tasks. The Standalone coref model is trained using only the coreference component of our joint architecture (eq. (2)–(4)), while the EL model is based only on the linking component (eq. (6)).

| Dataset | # Linked clusters | # NIL clusters | Linked mentions | # NIL mentions |
|---|---|---|---|---|
| DWIE | 11,967 | 9,935 | 28,482 | 14,891 |
| AIDA | 16,673 | - | 27,817 | 7,112 |
| AIDA$^+$ | 16,775 | 4,284 | 28,813 | 6,116 |

Table 1: Datasets statistics.

As performance metrics, for coreference resolution we calculate the average-F1 score of commonly used MUC (Vilain et al., 1995), B$^3$ (Bagga and Baldwin, 1998) and CEAF$_\text{e}$ (Luo, 2005) metrics as implemented by Pradhan et al. (2014). For EL, we use (i) *mention*-level F1 score (EL$_\text{m}$), and (ii) *cluster*-level *hard* F1 score (EL$_\text{h}$) that counts a true positive only if both the coreference cluster (in terms of all its mention spans) and the entity link are correctly predicted. These EL metrics are executed in a *strong matching* setting that requires predicted spans to exactly match the boundaries of gold mentions. Furthermore, for EL we only report the performance on non-NIL mentions, leaving the study of NIL links for future work.

Our experiments will answer the following research questions: **(Q1)** How does performance of our joint coref+EL models compare to Standalone models? **(Q2)** Does jointly solving coreference resolution and EL enable more coherent EL predictions? **(Q3)** How do our joint models perform on hard cases where some individual entity mentions do not have the correct candidate?

## 4 Results

Table 2 shows the results of our compared models for EL and coreference resolution tasks. Answering **(Q1)**, we observe a general improvement in performance of our coref+EL joint models (Local and Global) compared to Standalone on the EL task. Furthermore, this difference is bigger when using our cluster-level *hard* metrics. This also answers **(Q2)** by indicating that the joint models tend to produce more coherent cluster-based predictions. To make this more explicit, Table 3 compares the accuracy for singleton clusters (i.e., clusters composed by a single entity mention), denoted as $S$, to that of clusters composed by multiple mentions, denoted

[5] We use Wikipedia version 20200701.

[6] https://github.com/dalab/end2end_neural_el

| Setup | DWIE | | | $\text{AIDA}_a^+$ | | | $\text{AIDA}_b^+$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\text{EL}_\text{m}$ | $\text{EL}_\text{h}$ | Coref | $\text{EL}_\text{m}$ | $\text{EL}_\text{h}$ | Coref | $\text{EL}_\text{m}$ | $\text{EL}_\text{h}$ | Coref |
| Standalone | 88.7±0.1 | 78.4±0.2 | 94.5±0.1 | 86.2±0.4 | 80.7±0.5 | 93.8±0.1 | 79.1±0.3 | 74.0±0.3 | 91.5±0.3 |
| Local | 90.5±0.4 | 83.4±0.4 | 94.4±0.2 | 87.5±0.2 | 83.1±0.2 | 94.7±0.1 | **79.9±0.4** | 75.8±0.3 | **92.3±0.1** |
| Global | **90.7±0.3** | **83.9**±0.5 | **94.7**±0.2 | **87.6±0.2** | **83.7**±0.3 | **95.1**±0.1 | 79.6±0.4 | **76.0**±0.4 | 92.2±0.2 |

Table 2: Experimental results (F1 scores defined in §3) using the Standalone coreference and EL models compared to our joint architectures (Local and Global), on DWIE and AIDA$^+$ datasets.

| Setup | DWIE | | $\text{AIDA}_a^+$ | | $\text{AIDA}_b^+$ | |
|---|---|---|---|---|---|---|
| | $S$ | $M$ | $S$ | $M$ | $S$ | $M$ |
| Standalone | 80.4 | 69.5 | 82.9 | 70.7 | 77.0 | 57.0 |
| Local | **82.6** | 78.6 | 84.9 | 74.8 | **79.8** | 61.4 |
| Global | **82.6** | **80.0** | **85.1** | **76.8** | 79.3 | **63.0** |

Table 3: Cluster-based accuracy of link prediction on singletons ($S$) and clusters of multiple mentions ($M$).

| Setup | DWIE | $\text{AIDA}_a^+$ | $\text{AIDA}_b^+$ |
|---|---|---|---|
| Standalone | 0.0 | 0.0 | 0.0 |
| Local | 41.7 | 27.4 | 26.9 |
| Global | **57.6** | **50.2** | **29.7** |

Table 4: EL accuracy for corner case mentions where the correct entity is not in the mention's candidate list.

as $M$. We observe that the difference in performance between our joint models and Standalone is bigger on $M$ clusters (with a consistent superiority of Global), indicating that our approach indeed produces more coherent predictions for mentions that refer to the same concept. Further analysis reveals that this difference in performance is even higher for a more complex scenario where the clusters contain mentions with different surface forms (not shown in the table).

In order to tackle research question **(Q3)**, we study the accuracy of our models on the important corner case that involves mentions without correct entity in their candidate lists. This is illustrated in Table 4, which focuses on such mentions in clusters where at least one mention contains the correct entity in its candidate list. As expected, the Standalone model cannot link such mentions, as it is limited to the local candidate list. In contrast, both our joint approaches can solve some of these cases by using the correct candidates from other mentions in the cluster, with a superior performance of our Global model compared to the Local one.

## 5 Related Work

**Entity Linking:** Related work in entity linking (EL) tackles the document-level linking coherence by exploring relations between entities (Kolitsas et al., 2018; Yang et al., 2019; Le and Titov, 2019), or entities and mentions (Le and Titov, 2018). More recently, contextual BERT-driven (Devlin et al., 2019) language models have been used for the EL task (Broscheit, 2019; De Cao et al., 2020, 2021; Yamada et al., 2020) by jointly embedding mentions and entities. In contrast, we explore a cluster-based EL approach where the coherence is achieved on *coreferent* entity mentions level.
**Coreference Resolution:** Span-based antecedent-ranking coreference resolution (Lee et al., 2017, 2018) has seen a recent boost by using SpanBERT representations (Xu and Choi, 2020; Joshi et al., 2020; Wu et al., 2020). We extend this approach in our Local joint coref+EL architecture. Furthermore, we rely on Kirchhoff's Matrix Tree Theorem (Koo et al., 2007; Tutte, 1984) to efficiently train a more expressive spanning tree-based Global method.
**Joint EL+Coref:** Fahrni and Strube (2012) introduce a more expensive rule-based Integer Linear Programming component to jointly predict coref and EL. Durrett and Klein (2014) jointly train coreference and entity linking without enforcing single-entity per cluster consistency. More recently, Angell et al. (2021); Agarwal et al. (2021) use additional logic to achieve consistent cluster-level entity linking. In contrast, our proposed approach constrains the space of the predicted spanning trees on a structural level (see Fig. 1).

## 6 Conclusion

We propose two end-to-end models to solve entity linking and coreference resolution tasks in a joint setting. Our joint architectures achieve superior performance compared to the standalone counterparts. Further analysis reveals that this boost in performance is driven by more coherent predictions on the level of mention clusters (linking to the same entity) and extended candidate entity coverage.

## References


Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2021. Entity linking and discovery via arborescence-based supervised clustering. *arXiv preprint arXiv:2109.01242*.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021)*, pages 2598–2608.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1998 International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference (LREC 1998)*, pages 563–566.

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 2019)*, pages 677–685.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021)*.

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 (NAACL-HLT 2019)*, pages 4171–4186.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics (TACL 2014)*, 2:477–490.

Sourav Dutta and Gerhard Weikum. 2015. C3EL: A joint model for cross-document co-reference resolution and entity linking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 846–856.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240.

Angela Fahrni and Michael Strube. 2012. Jointly disambiguating and clustering concepts and entities with markov logic. In *Proceedings of the 2012 International Conference on Computational Linguistics (COLING 2012)*, pages 815–832.

Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 289–299.

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL 2020)*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 5807–5812.

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

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 141–150.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 1595–1604.

Phong Le and Ivan Titov. 2019. Boosting entity linking performance by leveraging unlabeled documents. In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 1935–1945.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on*

*Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 188–197.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 687–692.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3036–3046.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)*, pages 25–32.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 30–35.

William Tutte. 1984. Graph theory. *Encyclopedia of Mathematics and its Applications*, 21.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 1995 Conference on Message understanding (MUC6, 1995)*, pages 45–52.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 6953–6963.

Liyan Xu and Jinho D Choi. 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 2020)*, pages 8527–8533.

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 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 250–259.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2020. Global entity disambiguation with pretrained contextualized embeddings of words and entities. *arXiv preprint arXiv:1909.00426*.

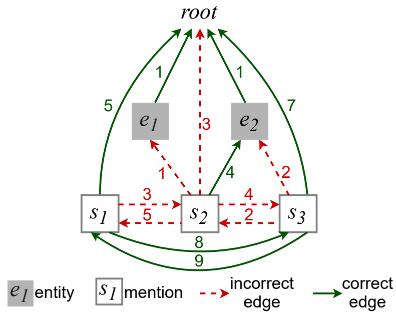


Figure 2: Illustrative graph example of Global model. The weights of the edges correspond to $\exp(\mathbf{\Phi}_{\text{cl}})$ (see eq. (11)).

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 International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 271–281.

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. DWIE: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 58(4):102563.

## A Step by Step Example of MTT Theorem

In this appendix we will provide a clarifying artificial example in order to walk the reader step by step through MTT (eq. (9)–(10)) applied in our Global approach. The graph of the example is illustrated in Fig. 2 and is composed by nodes representing $root$ ($r$), entities $e_1$ and $e_2$, and spans $s_1$, $s_2$ and $s_3$. The span $s_2$ is associated with candidate entity set $\{e_1, e_2\}$ (i.e., represented by edges from $s_2$ to $e_1$ and $e_2$), and $s_3$ with $\{e_2\}$ (i.e., represented by the edge from $s_3$ to $e_2$). The candidate entity set of $s_1$ is empty. The nodes are grouped in two ground truth clusters: NIL cluster $c_1 = \{s_1, s_2\}$, and linked cluster $c_2 = \{e_2, s_2\}$.

The exponential of weighted adjacency matrix[7] $\mathbf{\Phi}_{\text{cl}}$ of the presented example is:

$$\exp(\mathbf{\Phi}_{\text{cl}}) = \begin{array}{c} \\ r \\ e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{cccccc} r & e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 5 & 3 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 5 & 9 \\ 0 & 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 8 & 4 & 0 \end{bmatrix} \end{array}, \quad (11)$$

[7] For simplicity, the weights are small integers.

where the weights of incorrect edges are represented in red (i.e., red dashed edges in Fig. 2), the weights of the correct edges in green (i.e., green edges in Fig. 2), and the weights between disconnected nodes are set to 0.

In order to compute the *denominator* of the loss function in eq. (9), the Laplacian of the matrix in eq. (11) is calculated as described in eq. (10), and the row and column corresponding to root $r$ removed (i.e., the *minor* $\mathbf{L}_r$ with respect to the root):

$$\mathbf{L}_r = \begin{array}{c c} & \begin{array}{ccccc} e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{array}{c} e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} & \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & -4 & -2 \\ 0 & 0 & 16 & -5 & -9 \\ 0 & 0 & -3 & 17 & -2 \\ 0 & 0 & -8 & -4 & 20 \end{bmatrix} \end{array}. \tag{12}$$

Following Kirchhoff's Matrix Tree Theorem (Koo et al., 2007; Tutte, 1984), the determinant of $\mathbf{L}_r$ equals to the sum of the weights of all possible spanning trees of the graph represented in Fig. 2:

$$\det(\mathbf{L}_r) = 3600 = \sum_{t \in \mathcal{T}_{all}} \exp\big(\Phi_{\text{tr}}(t)\big). \tag{13}$$

In order to compute the *numerator* of the loss function in eq. (9) (i.e., the sum of the weights of the spanning trees of ground truth clusters), we first mask out (set to zero) all the weights assigned to incorrect edges:

$$\exp(\mathbf{\Phi}_{\text{cl}})' = \begin{array}{c c} & \begin{array}{cccccc} r & e_1 & e_2 & s_1 & s_2 & s_3 \end{array} \\ \begin{array}{c} r \\ e_1 \\ e_2 \\ s_1 \\ s_2 \\ s_3 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 5 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 \end{bmatrix} \end{array} \tag{14}$$

Next, the *modified Laplacian* (i.e., Laplacian with the first row replaced by root $r$ selection weights) $\hat{\mathbf{L}}$ is calculated for both clusters $c_1$ and $c_2$:

$$\hat{\mathbf{L}}_{c_1} = \begin{array}{c c} & \begin{array}{cc} s_1 & s_3 \end{array} \\ \begin{array}{c} r \\ s_3 \end{array} & \begin{bmatrix} 5 & 7 \\ -8 & 9 \end{bmatrix} \end{array} \tag{15}$$

$$\hat{\mathbf{L}}_{c_2} = \begin{array}{c c} & \begin{array}{cc} e_2 & s_2 \end{array} \\ \begin{array}{c} r \\ s_2 \end{array} & \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \end{array} \tag{16}$$

The determinants of $\hat{\mathbf{L}}_{c_1}$ and $\hat{\mathbf{L}}_{c_2}$ equal to the sum of the weights of all spanning trees connecting the nodes in clusters $c_1$ and $c_2$ respectively:

$$\det(\hat{\mathbf{L}}_{c_1}) = 101 = \sum_{t \in \mathcal{T}_{c_1}} \exp\big(\Phi_{\text{tr}}(t)\big) \tag{17}$$

$$\det(\hat{\mathbf{L}}_{c_2}) = 4 = \sum_{t \in \mathcal{T}_{c_2}} \exp\big(\Phi_{\text{tr}}(t)\big) \tag{18}$$

Finally, in order to calculate the final loss, we replace the obtained results in eqs. (13), (17), and (18) in the loss function of eq. (9):

$$\mathcal{L} = -\log \frac{101 * 4}{3600}. \tag{19}$$

*Note*: strictly speaking, there are *three* clusters rooted in *root* in the graph of Fig. 2, the third one being $c_3 = \{e_1\}$, whose exponential weight is 1 by definition of $\Phi_{\text{cl}}(r, e_j) = 0$ (see §2.2), and has no impact in calculation of the loss function in eq. (19).