# Supplementary Material for Labeling Gaps Between Words: Recognizing Overlapping Mentions with Mention Separators

**Aldrian Obaja Muis** and **Wei Lu**
Singapore University of Technology and Design
{aldrian_muis,luwei}@sutd.edu.sg

## Abstract

This is the supplementary material for "Labeling Gaps Between Words: Recognizing Overlapping Mentions with Mention Separators" (Muis and Lu, 2017). This material explains in more depth the issue of spurious structures and also the experiments settings.

## 1 Details on Spurious Structures

About mention hypergraph, we remarked in Section 3.1 that the normalization term calculated by the forward-backward algorithm includes spurious structures, which are structures that are not part of the true normalization term. This section shows in more details how this is the case using some examples.

Consider the simplified mention hypergraph as shown in Figure 1 (top left) consisting of three words and where the possible edges have been restricted to what are shown in the figure. Also, let A, B, C, D, E, F respectively denote the edges $\mathbf{T}^1 \rightarrow (\mathbf{I}^1)$, $\mathbf{I}^0 \rightarrow (\mathbf{I}^1)$, $\mathbf{I}^1 \rightarrow (\mathbf{I}^2)$, $\mathbf{I}^1 \rightarrow (\mathbf{X})$, $\mathbf{I}^1 \rightarrow (\mathbf{I}^2, \mathbf{X})$, and $\mathbf{I}^2 \rightarrow (\mathbf{X})$ as shown in Figure 1 (left). Further assume that features are only defined on these labeled edges.

Recall that in graphical models, any prediction by the model forms a *(hyper-)path* from the root node (here $\mathbf{A}^0$) to the leaf node ($\mathbf{X}$), which means each node other than the leaf node has exactly one outgoing (hyper-)edge. Now, notice that there are only three possible paths here, one for each of the three (hyper-)edges coming out from the node $\mathbf{I}^1$ associated with the word "Apache". See the top right, bottom left, and bottom right of Figure 1 for the visualization.

Now recall that in mention hypergraph, each node is assigned a certain set of mention combinations which the node represents, as defined by the subgraph from this node to the leaf node $\mathbf{X}$. In particular, the node $\mathbf{I}^1$ encodes three partial mentions (partial because the start of the mentions are undefined yet, as they are defined by the edge from $\mathbf{T}$-nodes to $\mathbf{I}$-nodes): {"Apache helicopter"}, {"Apache"}, and {"Apache","Apache helicopter"}, as shown in Figure 2. Similarly, we can see the subgraphs rooted at nodes $\mathbf{A}^1$ and $\mathbf{E}^0$ in Figure 3 and 4, respectively.

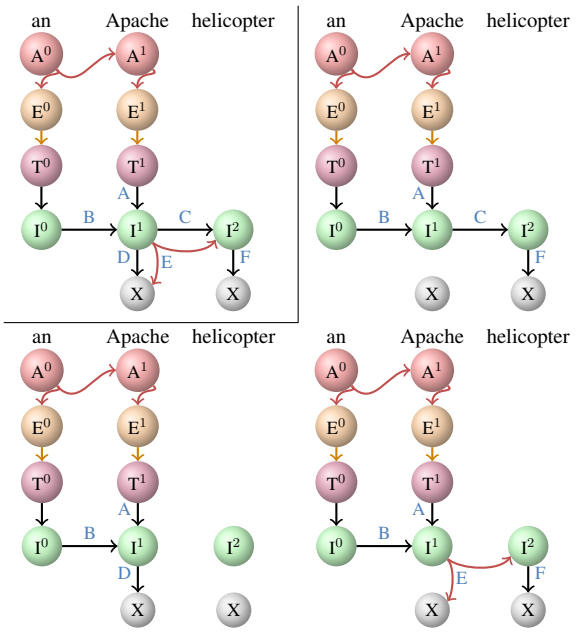


Figure 1: (top left) A simplified example of a mention hypergraph with restricted edges. (others) The three possible (hyper-)paths from the root node to the leaf node.

Now, the node $\mathbf{A}^0$ includes the 9 possible entity combinations which are the results of taking all possible combinations in $\mathbf{A}^1$ and $\mathbf{E}^0$. The list of mention combinations and the respective paths used to represent each of these combinations is as follows:

a. A-C-F and B-C-F (*Apache helicopter*, *an Apache helicopter*)

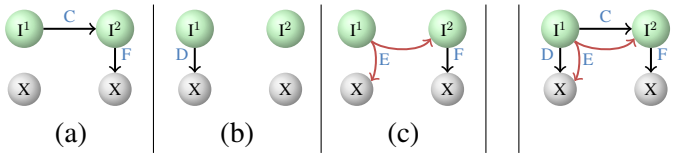


Figure 2: (left) The graphical representation of all 3 mention combinations represented by the node $\mathbf{I}_1$. (right) The full graph rooted at $\mathbf{I}^1$.

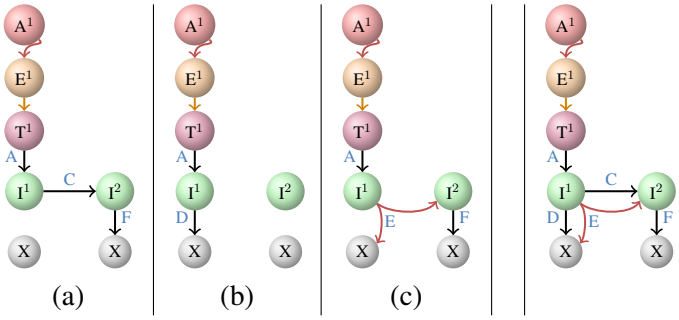


Figure 3: (left) The graphical representation of all 3 mention combinations represented by the node $\mathbf{A}_1$. (right) The full graph rooted at $\mathbf{A}^1$.

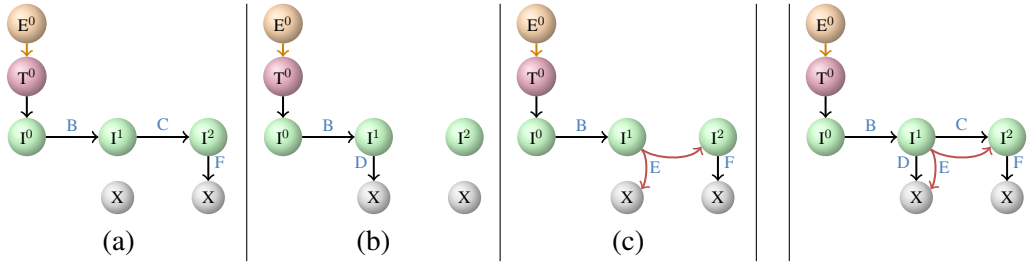


Figure 4: (left) The graphical representation of all 3 mention combinations represented by the node $\mathbf{E}_0$. (right) The full graph rooted at $\mathbf{E}^0$.

b. A-C-F and B-D (*Apache helicopter*, *an Apache*)
c. A-C-F and B-E-F (*Apache helicopter*, *an Apache helicopter*, *an Apache*)
d. A-D and B-C-F (*Apache*, *an Apache helicopter*)
e. A-D and B-D (*Apache*, *an Apache*)
f. A-D and B-E-F (*Apache*, *an Apache helicopter*, *an Apache*)
g. A-E-F and B-C-F (*Apache helicopter*, *Apache*, *an Apache helicopter*)
h. A-E-F and B-D (*Apache helicopter*, *Apache*, *an Apache*)
i. A-E-F and B-E-F (*Apache helicopter*, *Apache*, *an Apache helicopter*, *an Apache*)

See Figure 5 for a graphical representation of those 9 mention combinations. We display some nodes twice to highlight the different paths representing each distinct mention, a result of considering the mentions in $\mathbf{A}^1$ and $\mathbf{E}^0$ independently. Notice that this differing paths exist because the node $\mathbf{I}^1$ has two incoming edges in the path from $\mathbf{A}^0$ to $\mathbf{I}^1$. Further note that these are only the graphical representations of the mention combinations represented by the root node, as defined by the scoring of the structures in the objective function.

The crucial observation is that out of those nine mention combinations, only three of them form valid paths of the original graph in Figure 1, namely: (a), (e), and (i). The other six mention combinations will never be predicted by the model, as they do not form paths found in the full graph. For example, in (b), the node $\mathbf{I}^1$ has two outgoing edges: C and D, and so this graph does not form a path.

Those six mention combinations which are part of the structures represented by the root node $\mathbf{A}^0$ which do not form valid paths are the spurious structures, as during training they are calculated as part of the normalization term, but the model can not output any of those as a prediction, since they do not form a path.

In essence, the normalization term fails to take into account the restriction of forming a path when

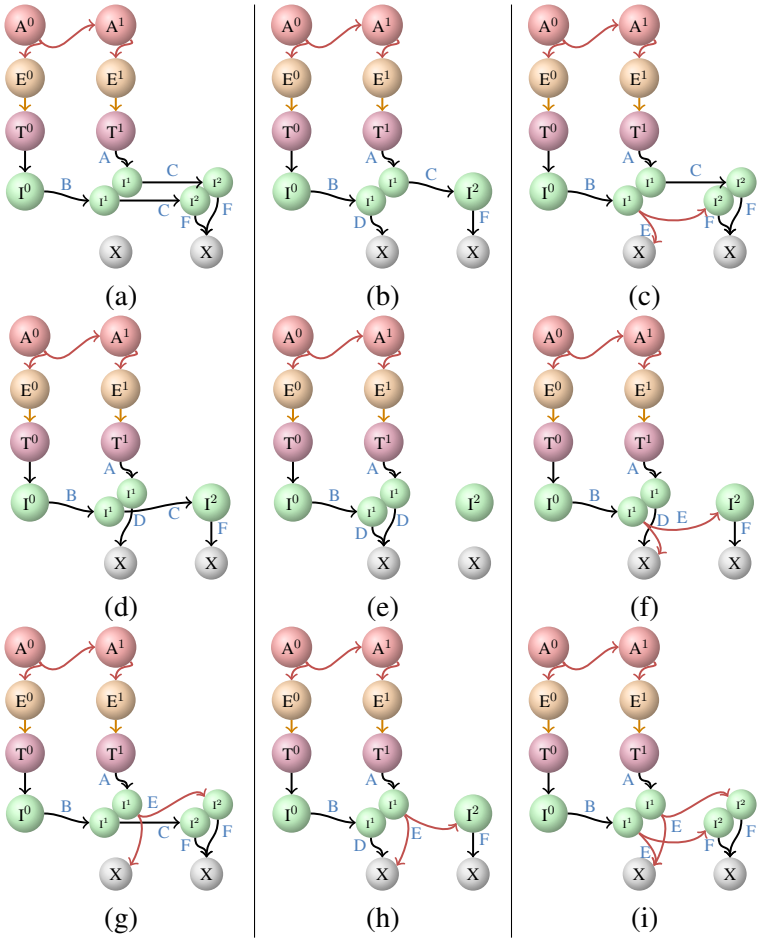


Figure 5: The graphical representation of all 9 mention combinations represented by node $\mathbf{A}^0$. See main text for details.

calculating the scores of all possible paths because it is considering the two sub-paths from $\mathbf{A}^1$ and $\mathbf{E}^0$ independently, which cannot capture the restriction that the node $\mathbf{I}^1$ should have only one outgoing edge. This is what causing the spurious structures issue in mention hypergraph.

As a final remark, note that, depending on the heuristics used, when the model outputs structure (i) as its prediction, we can still interpret that structure to mean any of the six mention combinations. More technically, this means that the spurious structures do not affect the interpretation process, but they do affect how the objective function is calculated, which in turns affects how the learning process goes. And as can be seen from the experiments, removing these spurious structures from the objective function indeed improves the entity recognition ability of the model.

## 2 Mention Separators

This section will give more examples and illustrations on how mention separators can be used to encode any overlapping mentions in a sentence, following the description at Section 4 in the main paper.

Suppose we want to encode the phrase "the IL2 regulatory region" containing two DNA mentions: "IL2" and "regulatory region" (example taken from GENIA dataset). In Figure 6, we start in step (1) by taking note of the possible starting, continuing, or ending marker in the gaps between the words. Then in step (2) we process the mention "IL2", marking the starting and ending marker before and after the mention accordingly. In step (3) we process the mention "IL2 regulatory region", marking the starting, continuing, and ending marker accordingly. Note that here it shares the starting marker as the previous mention. After all mentions have been processed, at step (4) we convert the combination of markers at each gap to the corresponding mention separator. Finally, we model the resulting sequence of mention separators into the EDGE-based model, selecting the edge corresponding to the mention separators.

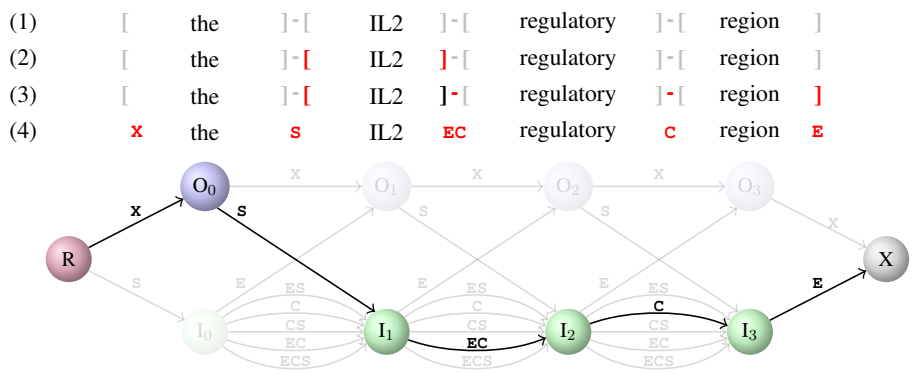


Figure 6: Example on how to encode mentions using mention separators with the EDGE-based model.

The decoding process will be the reverse of the encoding process, while using some heuristics to interpret the mention separator sequence. Continuing the example, given the path in the EDGE-based model, we extract the sequence of mention separators, and then convert them into the markers at each gap, resulting in the one we see at step (3). Now, we need to interpret this sequence to find out what are the mentions encoded by this sequence. First we note that since the end marker after the word "region" is active, that means there is a mention ending at that word. And since the only active starting marker is the one before the word "IL2", it must be the case that "IL2 regulatory region" is *one* of the mentions encoded by this sequence. This mention already explains the presence of most active markers, except the end marker after the word "IL2". This means there is another mention ending with the word "IL2". And again, since there is only one active starting marker, we conclude that "IL2" is another mention encoded by the sequence. Finally, we note that these two mentions already explain the mention separator sequence (i.e., these two mentions, when encoded, will result in the same mention separator sequence that we have), and so we end the interpretation process.

Note that the example in Figure 6 shows only the graph for recognizing one type. In the full model, there will be multiple chains, one for each type. An example of the full model can be seen in Figure 7.

### 2.1 Relation to Previous Work

We also want to remark that the number of paths in our multigraph-based model can be calculated in the same way as how we previously calculated the number of canonical encoding for discontiguous mention recognition model, which is based on the mention hypergraph, using a transition matrix (Muis and Lu, 2016). In fact, the number of edges between two states in our multigraph-based model reflects the numbers in the transition matrix, and so this multigraph-based model can be seen as a model that utilizes the canonical structures found in the mention hypergraph model. This means the way we use multigraph to represent the overlapping mentions as modeled by the mention hypergraph model theoretically can also be applied to the discontiguous mention model.

However, the number of edges in the multigraph representation will explode. To illustrate, in mention hypergraph, since there is only 1 node per word (excluding the **A**, **E**, and **T** nodes), the multigraph-based model requires only $2^1 = 2$ states with $2^3 = 8$ edges per word, while in the discontiguous mention model supporting three components, since there are 5 nodes per word ($\mathbf{B}_0$, $\mathbf{O}_1$, $\mathbf{B}_1$, $\mathbf{O}_2$, and $\mathbf{B}_2$), the multigraph-based model will require $2^5 = 32$ nodes, with average number of edges per word being $2^{13} = 8192$, which makes it much slower than the hypergraph-based counterpart with only 25 edges per word. So we can say that our multigraph-based approach trades off the speed in mention hypergraph with a higher $F_1$-score.

## 3 Features

For ACE datasets we used these features:

1. Words and POS tags (with window of 3 words to the left and right of current word)
2. Words and POS tags n-gram (up to length 4 containing current word)
3. Bag-of-word (with window of 5 words to the left and to the right of current word)
4. Orthographic (following Lu and Roth (2015))
5. Parent node type

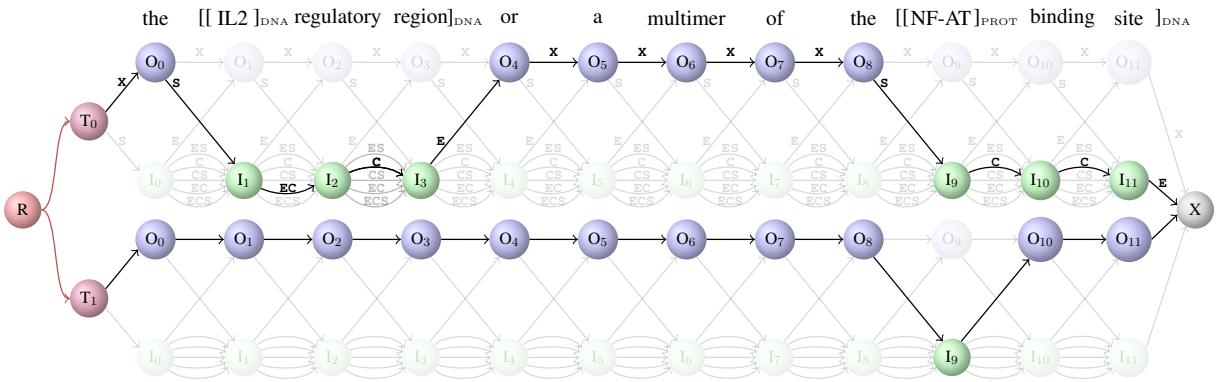


Figure 7: The multigraph model with two chains representing two types: DNA and PROT.

| | CoNLL2003-dev | | | CoNLL2003-test | | | CoNLL2003-dev ($F$ optimized) | | | CoNLL2003-test ($F$ optimized) | | | $w/s$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $P$ | $R$ | $F$ | $P$ | $R$ | $F$ | $P$ | $R$ | $F$ | $P$ | $R$ | $F$ | |
| LCRF (single) | 90.0 | 88.9 | **89.5** | 84.2 | 83.6 | **83.9** | 90.1 | 88.9 | **89.5** | 84.3 | 83.4 | 83.8 | 148.6 |
| LCRF (multiple) | 94.4 | 84.6 | **89.2** | 91.5 | 78.2 | **84.3** | 92.7 | 86.8 | **89.6** | 88.6 | 81.2 | **84.7** | 283.4 |
| Ratinov and Roth (2009) | - | - | 89.3 | - | - | 83.7 | - | - | - | - | - | - | - |
| Lu and Roth (2015) | 94.4 | 83.4 | 88.5 | 91.1 | 77.0 | 83.5 | 89.7 | 88.7 | 89.2 | 84.6 | 82.9 | 83.8 | 1169.7 |
| This work (STATE) | 94.2 | 84.7 | **89.2** | 91.1 | 78.2 | **84.2** | 91.2 | 88.1 | **89.6** | 86.3 | 82.4 | **84.3** | 116.3 |
| This work (EDGE) | 94.5 | 84.6 | **89.3** | 91.3 | 78.2 | **84.3** | 93.3 | 86.5 | **89.8** | 89.2 | 80.4 | **84.6** | 554.0 |

Table 1: Complete results on CoNLL-2003.

For GENIA we used these features:

1. Words and POS tags (with window of 2 words to the left and right of current word)
2. Words and POS tags n-gram (up to length 4 containing current word)
3. Bag-of-word (with window of 5 words to the left and right of current word)
4. Brown clusters with window of 1 word to the left and right of current word (using 100 or 1000 clusters built from training data only)
5. Word shape (Finkel and Manning (2009))
6. Prefixes and suffixes of current word (up to length 6)
7. Edge type

For CoNLL 2003 we used these features:

1. Words and POS tags (with window of 2 words to the left and right of current word)
2. Words and POS tags n-gram (up to length 4 containing current word)
3. Bag-of-word (with window of 5 words to the left and right of current word)
4. Word shape (with window of 2 words to the left and right of current word)
5. Prefixes and suffixes of current word (up to length 5)
6. Orthographic (following Lu and Roth (2015))
7. Edge type

The mention penalty feature was added to all models by assigning it to the edges which have the semantics of starting a new entity, similar to how it was defined originally in Lu and Roth (2015). More specifically, for linear-chain CRF models we add the mention penalty feature to all incoming edges of the **B** and **U** nodes. For our STATE-based model we add it to all incoming edges of the nodes that includes `S`. For our EDGE-based model we add it to the incoming edges of the **I** nodes.

## 4 GENIA Preprocessing

For GENIA, we used GENIAcorpus3.02p that comes with POS tags for each word (Tateisi and Tsujii, 2004). Similar to the problem faced by Finkel and Manning (2009) on JNLPBA dataset, we also find tokenization issues in this corpus. As described by Tateisi and Tsujii (2004), when a hyphenated word such as *IL-2-induced* is partially annotated as an entity (in this case *IL-2*), the POS annotation corpus splits it into two tokens, which when done in test set will leak some information about the presence of entity. Unlike Finkel and Manning (2009) which tried to match the tokenization during testing, we simply further split all tokens at some punctuations (those matching the regular expression `[-/,.+]`), while keeping the information that they all originally come from the same word. This has the advantage of simplifying the tokenization procedure, although it makes the task slightly more difficult due to the higher num-

| | | % | LCRF (single) | | | LCRF (multiple) | | | Lu and Roth (2015) | | | This work (STATE) | | | This work (EDGE) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ | $P$ | $R$ | $F_1$ |
| ACE-2004 | O | 42 | 65.6 | 40.9 | 50.4 | 69.6 | 50.4 | 58.5 | 72.5 | 52.4 | 60.8 | 72.2 | 55.1 | 62.5 | 72.1 | 55.3 | 62.6 |
| | Ø | 58 | 67.4 | 66.6 | 67.0 | 70.5 | 68.2 | 69.4 | 72.5 | 65.0 | 68.6 | 74.2 | 65.4 | 69.5 | 74.1 | 65.5 | 69.5 |
| ACE-2005 | O | 32 | 64.6 | 43.4 | 51.9 | 68.5 | 49.3 | 57.4 | 68.1 | 52.6 | 59.4 | 68.6 | 54.7 | 60.8 | 70.4 | 55.0 | 61.8 |
| | Ø | 68 | 60.4 | 62.7 | 61.6 | 64.1 | 65.3 | 64.7 | 64.1 | 65.1 | 64.6 | 64.9 | 62.7 | 63.8 | 67.2 | 63.4 | 65.2 |
| GENIA | O | 24 | 78.3 | 52.6 | 62.9 | 78.0 | 59.2 | 67.3 | 76.3 | 60.8 | 67.7 | 77.0 | 60.0 | 67.5 | 76.5 | 60.3 | 67.4 |
| | Ø | 76 | 76.6 | 70.6 | 73.4 | 74.7 | 70.7 | 72.7 | 73.1 | 70.7 | 71.9 | 75.2 | 70.6 | 72.8 | 74.8 | 71.3 | 73.0 |

Table 2: Results on different types of sentences.

ber of tokens.

Also, to handle the discontiguous entities present in GENIA dataset (mainly due to coordinated entities involving ellipsis), following the approach used by the JNLPBA shared task organizer (Kim et al., 2004), we consider a group of coordinated entities as one structure. For example, in "*... the [T- and B-lymphocytes] count in ...*", the entities "*T-lymphocytes*" and "*B-lymphocytes*" are annotated as one structure "*T- and B-lymphocytes*".

## 5 Results on CoNLL-2003

Table 1 shows the full result of the experiments on CoNLL-2003 dataset, which includes the result in CoNLL-2003 development set, and also the results after optimizing the $F_1$ score. We see that since the precision and recall in LCRF model is already balanced, optimizing the $F_1$ score does not improve much, and even slightly decrease the result in the test set. We do see, however, some slight improvements in other models.

## 6 Results on Overlapping and Non-overlapping Sentences

Table 2 shows the complete scores of each model on the two subsets of the test set: the overlapping ones (O) and the non-overlapping ones (Ø). We can see that the edge-based model is quite robust to the amount of overlapping mentions in the dataset. For example, in GENIA dataset where the proportion of overlapping mentions is lower compared to the ACE datasets, it still achieves good results both in the overlapping part and the non-overlapping part.

## 7 Hyperparameter

For each model, we tuned the $l_2$-regularization coefficient $\lambda$ from the values {0.0, 0.001, 0.01, 0.1, 1.0}. And for GENIA we additionally tuned the number of Brown clusters used from the values {100, 1000}. Table 3 lists the optimal $\lambda$ for each dataset and model. For GENIA, the optimal Brown cluster size was found to be 1000, except for 'This work (STATE)', where the best cluster size is found to be 100.

| | ACE'04 | ACE'05 | GENIA | CoNLL |
|---|---|---|---|---|
| LCRF (single) | 0.1 | 0.01 | 0.1 | 0.001 |
| LCRF (multiple) | 0.001 | 0.0 | 1.0 | 0.001 |
| Lu and Roth (2015) | 0.001 | 0.0 | 1.0 | 0.01 |
| This work (STATE) | 0.0 | 0.001 | 1.0 | 0.001 |
| This work (EDGE) | 0.001 | 0.001 | 1.0 | 0.001 |

Table 3: The value of $l_2$ regularization parameter that gives the best result in development set.

## References


Jenny Rose Finkel and Christopher D. Manning. 2009. Nested Named Entity Recognition. In *Proc. of EMNLP*, page 141.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the Bio-entity Recognition Task at JNLPBA. In *Proc. of JNLPBA*, page 70.

Wei Lu and Dan Roth. 2015. Joint Mention Extraction and Classification with Mention Hypergraphs. In *Proc. of EMNLP*, pages 857–867. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2016. Learning to Recognize Discontiguous Entities. In *Proc. of EMNLP*, pages 75–84, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling Gaps Between Words: Recognizing Overlapping Mentions with Mention Separators. In *Proc. of EMNLP*, Copenhagen, Denmark. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proc. of CoNLL*, page 147. Association for Computational Linguistics.

Yuka Tateisi and Jun'ichi Tsujii. 2004. Part-of-Speech Annotation of Biology Research Abstracts. In *Proc. of LREC*, pages 1267–1270.