

# A Fast and Sound Tagging Method for Discontinuous Named-Entity Recognition

Caio Corro

INSA Rennes, IRISA, Inria, CNRS, Université de Rennes  
caio.corro@irisa.fr

## Abstract

We introduce a novel tagging scheme for discontinuous named entity recognition based on an explicit description of the inner structure of discontinuous mentions. We rely on a weighted finite state automaton for both marginal and maximum *a posteriori* inference. As such, our method is sound in the sense that (1) well-formedness of predicted tag sequences is ensured via the automaton structure and (2) there is an unambiguous mapping between well-formed sequences of tags and (discontinuous) mentions. We evaluate our approach on three English datasets in the biomedical domain, and report comparable results to state-of-the-art while having a way simpler and faster model.

## 1 Introduction

Named-entity recognition (NER) is a fundamental natural language processing (NLP) task that aims at identifying mentions of named entities in texts. These mentions may for example refer to persons, organizations, locations or even dates, among others (Grishman and Sundheim, 1996; Chinchor and Robinson, 1998). Over the years, this task has been extensively studied by the community, with contributions including decoding algorithms, neural network architectures, loss functions and methods for learning in different data availability situations, *inter alia*.

There exists several variants of the NER problem, among which the most studied are *flat* NER and *nested* NER. The most common method for the flat case is BIO tagging (Ramshaw and Marcus, 1995), where each word in a sentence is tagged depending on whether it is the beginning of a mention (B), inside a mention (I) or outside a mention (O).<sup>1</sup> This tagging scheme can be augmented to disambiguate types, *e.g.* BLOC and BPER. An important benefit of BIO tagging is that prediction has a linear time-

complexity in the input length<sup>2</sup> using the Viterbi algorithm (Forney, 1973), contrary to concurrent approaches like semi-Markov models that have a quadratic time-complexity (Janssen and Limmios, 1999; Ge, 2002; Sarawagi and Cohen, 2004).

A less studied task is *discontinuous* NER, where mentions are allowed to span discontinuous sequences of words. This problem is especially important for biomedical NLP. For example, pharmacovigilance aims to detect adverse drug reactions after a product is distributed in the market via automatic analysis of medical reports or social media (Berlin et al., 2008; Coloma et al., 2013). Mentions of adverse drug reactions naturally occur in non-contiguous sequences, for example the sentence “The pain I was experiencing around the hipjoints was incredible” contains the mention “pain hipjoints” with a five word gap in the middle.

Several methods for discontinuous NER have been proposed in the literature, including transition models (Dai et al., 2020) and other structured prediction approaches (Wang et al., 2021; Fei et al., 2021; Li et al., 2022). Unfortunately, they are more costly than BIO tagging and require specialized neural network architectures. There have also been attempts to propose tagging schemes for discontinuous NER (Tang et al., 2013, 2018; Metke-Jimenez and Karimi, 2016; Muis and Lu, 2016), but they all exhibit *structural ambiguity* (see Section 5).

In this work, we propose a novel tagging scheme for discontinuous NER that exploits the inner structure of discontinuous mentions. Contrary to previous attempts, our approach is *sound* in the sense that: (1) there is no encoding ambiguity between sets of mentions and sequences of tags (*i.e.* there is a one-to-one mapping between the two representations); and (2) our prediction algorithm is con-

<sup>2</sup>It is quadratic in the number of tags, which depends on the number of possible mention types. However, types are not considered part of the input and are assumed to be fixed.

<sup>1</sup>See (Ratinov and Roth, 2009) for other variants.

strained to predict only well-formed sequences of tags (*i.e.* we can always reconstruct a set of mentions from a predicted tag sequence). To ensure well-formedness of predictions, we propose an algorithm based on inference in a weighted finite-state automaton. Using our approach, the time complexity of maximum *a posteriori* inference for prediction is linear in the length of the input. Moreover, our algorithm can be very efficiently implemented on GPU for batched inference (Argueta and Chiang, 2017; Rush, 2020).

Our contributions can be summarized as follows:

- We propose to decompose discontinuous mentions in a new two-layer representation;
- We propose a novel tagging scheme for this representation together with a linear-time tagging algorithm that ensures well-formedness of predictions;
- We explain how labels in the inner structures can be inferred during training when the information is not available in the data;
- We experiment on three English datasets and report competitive results while having a much faster model.

Our implementation is publicly available.<sup>3</sup> Importantly, our decoding algorithm and all our loss functions can be used as a drop-in replacements in any BIO tagger. As such, any future research in the BIO tagging field may also be evaluated on discontinuous NER at no extra cost.

## 2 Reduction to Word Tagging

In this section, we explain how we map discontinuous mentions into a two-layer representation that allows us to derive a new tagging scheme. Although this transformation is generic, for ease of exposition we illustrate it on the particular case of adverse drug reactions.

### 2.1 Inner Structure of Mentions

Discontinuous mentions of adverse drug reactions (ADR) and disorders in biomedical NER mainly result from two linguistic phenomena. Firstly, mentions may be expressed as the combination of two non-contiguous syntactic constituents, due to linguistic word order rules. In the following example of an ADR, the discontinuity

<sup>3</sup><https://github.com/FilippoC/disc-ner-tagging>

is caused by the verb position constraint in English:

(1)  $\frac{\text{ADR}}{\text{toes are painful}}$

Secondly, many languages allow alternative sentential structures for coordinations, including construction based on deletion operations. For example, consider the two following sentences:

(2)  $\frac{\text{ADR}}{\text{pain in arms and pain in shoulders}}$

(3)  $\frac{\text{ADR}}{\text{pain in arms and shoulders}}$

The repeated element is eliminated in the second one, leading to the presence of a discontinuous mention, a phenomenon called coordination reduction (Lakoff and Peters, 1969). Although the underlying linguistic structures are different, we will treat both cases in the same way.

**Change of representation.** In practice, discontinuous mentions exhibit an inner structure. For example, a discontinuous ADR can be decomposed into a *body part* and an *event*. As such, we propose to transform discontinuous mentions into a two-layer representation:

- Upper layers identify *sets of mentions*;
- Lower layers identify *typed components*.

We restrict the number of types for components to be equal to two. The previous example is converted as follows:

(4)  $\frac{\text{ADR(S)}}{\frac{\text{EVENT PART}}{\text{pain in arms}} \text{ and } \frac{\text{PART}}{\text{shoulders}}}$

Note that the two mentions do not explicitly appear in this new representation. Nevertheless, the opposite transformation is trivial: to rebuild all discontinuous mention in a discontinuous set, we simply take the Cartesian product between the two sets of typed components, *e.g.*

$$\underbrace{\{\text{pain in}\}}_{\text{Components typed EVENT}} \times \underbrace{\{\text{arms, shoulders}\}}_{\text{Components typed PART}} \mapsto \underbrace{\{\text{pain in arms, pain in shoulders}\}}_{\text{Reconstructed discontinuous mentions}}$$

Note that this can result in some of the mentions being continuous, as in Example (4).

One obvious issue is that component types are not annotated in datasets. We consider two solutions to tackle this challenge. First, we can use unsupervised and weakly-supervised learning

methods to infer component types during training, as explained in Section 4. Second, we can use component types to mark if they share the same type as the leftmost one, no matter whether they refer to a body part of an event. In this setting, Examples (1) and (3) are annotated as follows:

(5)  $\frac{\text{ADR(S)}}{\text{FIRST} \quad \text{OTHER}}$   
toes are painful

(6)  $\frac{\text{ADR(S)}}{\text{FIRST} \quad \text{OTHER} \quad \text{OTHER}}$   
pain in arms and shoulders

In other words, component types do not convey semantic information, only structural information.

**Continuous mentions.** There exists two forms of continuous mentions. First, continuous mentions that share one or more words with at least one other mention. In this case, we split the mention and we process it as described above. Second, there are continuous mentions that do not share any word with other mentions, see Example (2). In principle, we could also transform these mentions in the two layers representation. However, not only we lack information about component types but we do not even know where to split them! In Example (3), we know that “pain in arms” should be splitted into “pain in” and “arms” as the first two words are shared with another mention. But for the two continuous mentions in Example (2), we do not have such information. Therefore, in this case, we treat them as standard continuous ones.

**Nested NER.** Although Dai et al. (2020) suggested the use of nested NER models for discontinuous NER using a similar yet different representation, we argue that the two problems are different:

- The structures that we consider are not recursive, contrary to nested mentions, e.g. “[The president of [the United States of [America]]]”;
- The components are highly constrained, e.g. a set of ADRs must contain at least one body part and one event;
- The span of a set of mentions is fixed by its components: it begins (resp. ends) at the same word as its leftmost (resp. rightmost) component.

Therefore, we instead propose a tagging scheme tailored to discontinuous NER.

**Beyond the biomedical domain.** Our approach can be applied to other domains, e.g. we can transform the following mentions into our representation by differentiating first and last names:

(7)  $\frac{\text{PER}}{\text{Meg} \quad \text{and} \quad \text{Jack} \quad \text{White}}$   
 $\text{PER}$

Unfortunately, these discontinuities have not been annotated in standard datasets.<sup>4</sup>

## 2.2 Tagging Scheme

We now explain how we transform the two-layer structure into a sequence of tags. Without loss of generality, we assume that mentions are untyped in the original corpus, as practical datasets for discontinuous NER contain a single mention type.<sup>5</sup> Moreover, we define the component types as X and Y (e.g. *body part* and *event* in previous examples) to simplify notation and treat in a similar way semantic and structural component types.

Our approach requires 10 tags. First, the 3 tags CB, CI and O are used in a similar way to BIO tags. CB and CI are used to identify first and following words in a continuous mention, respectively. The tag O is used to mark words that are neither part of a continuous mention or in the span of a set of mentions. In Example (2), word “and” is tagged with O whereas in Example (3) it is *not* tagged with O. This is due to the fact that in the second example, after transformation into the two layers representation, the word “and” will appear inside a set of mentions, see Example (4).

Second, tags to identify set of mentions and their components are of the form \*-\* where:

- the left-hand side is used to identify the span of the set of mentions, and can therefore take values DB (first word of the span) and DI (other words of the span);
- the right-hand side is used to identify typed components, and can take values BX, IX, BY, IY and O.

The 7 tags used for discontinuous mentions are DB-BX, DB-BY, IB-BX, IB-BY, IB-IX, IB-IY

<sup>4</sup>Wang et al. (2023) automatically extracted coordination structures from syntactic structures. However, note that (1) the resulting dataset does not contain discontinuous mentions that we are interested in and (2) conjunction reduction cannot always be inferred from the syntactic structure (Lakoff and Peters, 1969; Lechner, 2000; Wilder, 2018).

<sup>5</sup>It is trivial to augment the set of tags with types if necessary, as done for standard BIO tagging.

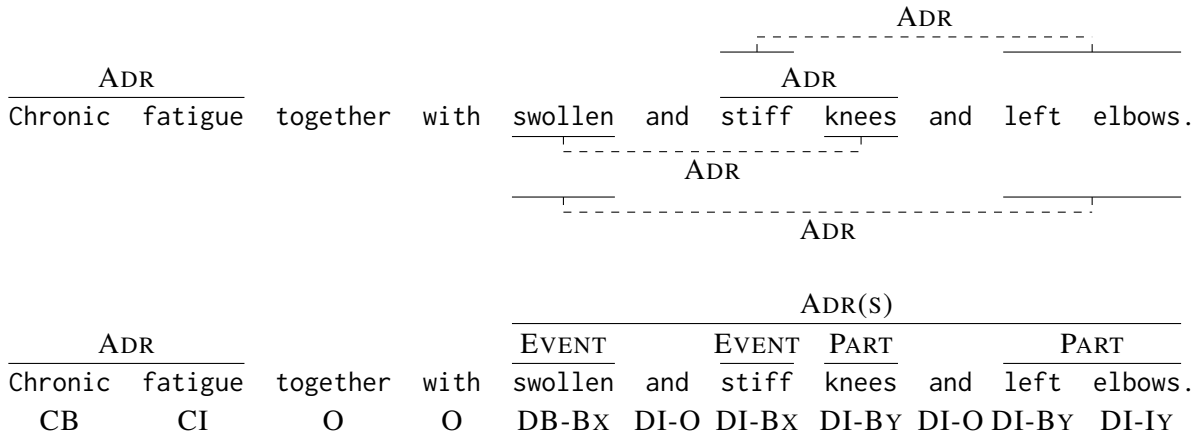


Figure 1: **(Top)** Sentence with its original annotation. It contains two continuous mentions (“Chronic fatigue” and “stiff knees”) and three discontinuous mentions (“swollen knees”, “swollen left elbows” and “stiff left elbows”). **(Bottom)** Sentence annotated with our two-layer representation and the associated tag sequence.

and IB-O. Note that the leftmost word in a set of mentions must also be the beginning of a component, so the following combinations *are not* part of the tagset: DB-IX, DB-IY and DB-O. Figure 1 shows an example of tag conversion.

Importantly, any sequence of tags is *well-formed* if and only if:

1. All CI tags are preceded by either BI or CI, as standard in BIO tagging;
2. All DI-\* tags must be preceded by either DB-\* or DI-\*;
3. All \*-IX tags must be preceded by either \*-BX or \*-IX (and similarly for the Y type);
4. A set of mentions must contain at least one component typed X and one typed Y, that is it must contain at least one word tagged with \*-BX and one with \*-BY.
5. A set of mentions must not yield a single continuous mention after reconstruction, *i.e.* the following sequence of tags is forbidden:
 

(8)  $\begin{array}{ccccccc} \text{some} & \text{pain} & \text{in} & \text{arms} & \text{and} & & \\ \text{O} & \text{DB-BX} & \text{DI-IX} & \text{DI-IY} & \text{O} & & \end{array}$

 as it would introduce ambiguity in the encoding of continuous mentions;
6. A discontinuous mention cannot end with tag DI-O, as this would result in the span of a set of mentions that do not end with the same word as its rightmost component.<sup>6</sup>

<sup>6</sup>The analogous constraint on the first word is implicitly enforced by the absence of a DB-O tag in the tagging scheme.

### 3 Decoding Algorithm

Without loss of generality, we assume all sentences have  $n$  words. Let  $T$  be the tagset,  $X$  be the set of sentences and  $Y$  the set of well-formed tag sequences. We represent a sequence of tags  $\mathbf{y} \in Y$  as a binary vector with  $n|T|$  entries, where each entry is associated with a tag and a word, *i.e.*  $\mathbf{y} \in \{0, 1\}^{n|T|}$ . If the value of an entry is 1 (resp. 0), the associated tag is assigned to the associated word (resp. not assigned). Note that  $Y \subset \{0, 1\}^{n|T|}$  is a strict subset of all such vectors, as each word must be assigned exactly one tag and that the resulting tag sequence must satisfy the constraints described in Section 2.2.

Let  $f_\theta : X \rightarrow \mathbb{R}^{n|T|}$  be a neural network parameterized by  $\theta$ . We define the probability of a tag sequence  $\mathbf{y} \in Y$  given the input  $\mathbf{x}$  as a Boltzmann-Gibbs distribution (or *softmax* over structures):

$$p_\theta(\mathbf{y}|\mathbf{x}) = \exp(\langle \mathbf{y}, f_\theta(\mathbf{x}) \rangle - A_Y(f_\theta(\mathbf{x}))),$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product and  $A_Y$  is the log-partition function ensuring that the distribution is correctly normalized:

$$A_Y(\mathbf{w}) = \log \sum_{\mathbf{y} \in Y} \exp \langle \mathbf{y}, \mathbf{w} \rangle. \quad (1)$$

Computing  $A_Y(\mathbf{w})$  is called *marginal inference* due to its link with marginal probabilities (Wainwright et al., 2008). Computing the most probable output is reduced to computing:

$$\hat{\mathbf{y}}_\theta(\mathbf{x}) = \arg \max_{\mathbf{y} \in Y} \langle \mathbf{y}, f_\theta(\mathbf{x}) \rangle, \quad (2)$$



called *maximum a posteriori (MAP) inference*.

In practice, we need to compute the term in Equation (1) for training the model and the term in Equation (2) for prediction. The difficulty stems from the restriction (in the sum and in the arg max search space) to the set of well-formed outputs  $Y$ . We follow a long tradition in NLP (Koskenniemi, 1990; Mohri et al., 1996; Karttunen et al., 1996; Kanthak and Ney, 2004; Tromble and Eisner, 2006; Rastogi et al., 2016; Lin et al., 2019; Papay et al., 2022, *inter alia*) and rely on a finite-state automaton to solve these inference problems.

### 3.1 Finite-State Automata

**Definitions.** Weighted Finite State Automata (WFSAs) are generalization of FSA (Eilenberg, 1974) that include weights on their transitions. Formally, a WFSAs over  $\mathbb{R}$  is a 5-tuple  $(\Sigma, Q, E, i, F)$  where:

- $\Sigma$  is a finite alphabet with  $\epsilon \notin \Sigma$ ;
- $Q$  is the set of states;
- $E \subseteq Q \times \Sigma^* \times \mathbb{R} \times Q$  is the set of weighted transitions, where  $(q, \sigma, w, r) \in E$  is a transition from state  $q$  to state  $r$  emitting symbol(s)  $\sigma$  with weight  $w$ ;
- $i \in Q$  is an initial state and  $F \subseteq Q$  are final states.

Symbol  $\epsilon$  is used for transitions that emit nothing. A WFSAs is  $\epsilon$ -free if there is no  $\epsilon$ -transition. A *valid path* is a path starting at  $i$  and ending at any state in  $F$ . A path emits a sequence of symbols, and has a weight equal to the sum of the transition weights it contains. The language of a WFSAs is the set of emissions along all possible valid paths.

**Algorithms.** Given an acyclic WFSAs, the path of maximum weight, Equation (2), and the log-sum-exp of all valid paths, Equation (1), can be computed using variants of the Viterbi algorithm (Forney, 1973) and the Forward algorithm (Baum, 1972), respectively. These algorithms are in fact identical, but defined over different semirings (Goodman, 1999): the tropical semiring for the Viterbi and the thermodynamic semiring (Marcolli and Thorngren, 2014) for the Forward. We refer to (Mohri, 2009, Section 3) for an in-depth introduction. The time complexity of both algorithms is  $\mathcal{O}(|E|)$  if a topological ordering of states is known.

**Application to sequence tagging.** We follow previous work and use the intersection of two WFSA to constraint tag sequences (Koskenniemi, 1990; Koskenniemi et al., 1992). The *grammar automaton*  $\mathcal{G} \triangleq (T, Q, E, i, F)$  is a cyclic WFSAs whose language is the set of all well-formed tag sequences (of any length). We assume  $\mathcal{G}$  is  $\epsilon$ -free and deterministic.<sup>7</sup> Without loss of generality, we fix all transition weights to 0. The *sentence automaton*  $\mathcal{S} \triangleq (T, Q', E', i', F')$  is an acyclic FSA that represents all possible (not necessarily valid) analyses for a given sentence of  $n$  words. States are  $Q' \triangleq \{0, \dots, n\}$  and transitions are:

$$E' \triangleq \{(i-1, t, w_{(i,t)}, i) \mid i \in \{1 \dots n\} \wedge t \in T\}$$

where  $w_{(i,t)}$  is the weight associated with tagging word at position  $i$  with tag  $t$ . Initial and final states are  $i' \triangleq 0$  and  $F' \triangleq \{n\}$ . This WFSAs contains  $n|T|$  transitions, and each transition correspond to tagging a given word with a given tag. By construction, it is always deterministic and  $\epsilon$ -free.

We denote  $G \cap S$  the intersection of  $G$  and  $S$  (Hopcroft et al., 2001, Section 4.2.1) composed of states  $Q'' \triangleq Q \times Q'$ , transitions

$$E'' \triangleq \left\{ \left( (i-1, p), t, w_{(i,t)}, (i, q) \right) \left| \begin{array}{l} i \in \{1 \dots n\} \wedge \\ (p, t, 0, q) \in E \end{array} \right. \right\},$$

initial state  $i'' \triangleq (i, i')$  and final states  $F'' \triangleq F \times F'$ . Then, all valid paths in  $G \cap S$  are well-formed sequences of tags for the input sentence of length  $n$ . We can then simply run the Viterbi or the Forward algorithm on  $G \cap S$  to compute Equations (1) and (2). Note that  $|E''| \propto n$ , therefore the time-complexity is linear in the number of words.

We refer the reader to (Tapanainen, 1997) for an introduction to this sequence tagging approach.

### 3.2 Grammar Automaton

The grammar automaton used to constraint prediction to well-formed sequences of tags is shown in Figure 2. We present the automaton with  $\epsilon$ -transition for the sake of clarity, but they can be removed. We omit weights as they are null. States 1 and 2 recognize valid sequences of CB, CI and O tags. Moreover, the structure of the WFSAs recognizing discontinuous mentions is symmetric: the left-hand (resp. right-hand) side recognizes discontinuous mentions whose leftmost component is

<sup>7</sup>Procedures to determinize and remove  $\epsilon$ -transitions can be found in Hopcroft et al. (2001, Section 2.3.5 and 2.5.5).

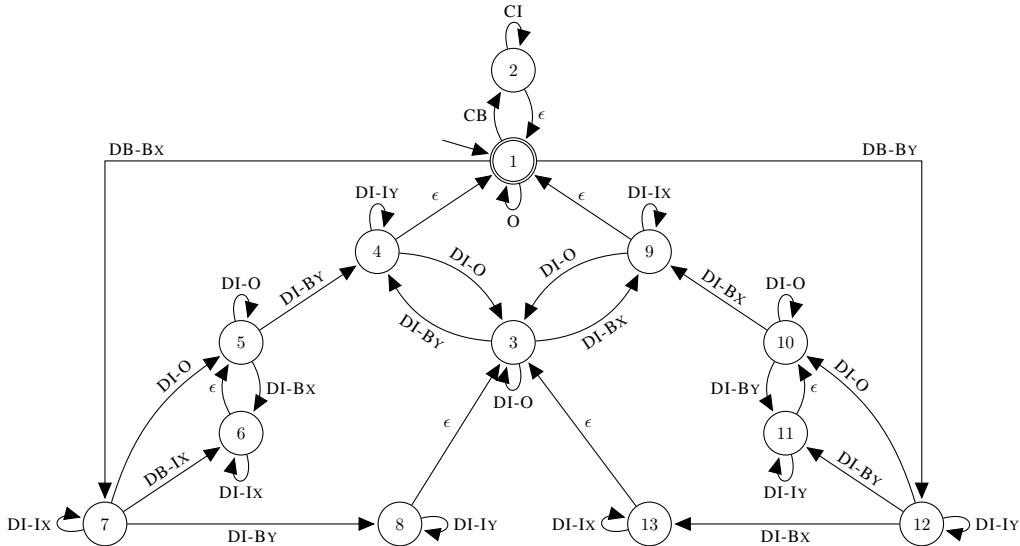


Figure 2: The grammar automaton we propose for discontinuous named-entity recognition.

typed  $X$  (resp.  $Y$ ). Therefore we present only the left-hand side.

Transition  $(1, DB-BX, 7)$  starts the recognition of a set of mentions whose leftmost component is typed  $X$ . The self-loop in state 7 recognizes following words of the first component. Next we need to check that the inner structure of the set of mentions is well-formed. On the one hand, states 5 and 6 allows to recognize following  $X$  components and  $DI-O$  tags, until recognition of the first  $Y$  component via transition  $(5, DI-BY, 4)$ . On the other hand, transition  $(7, DB-BX, 8)$  starts the recognition of an component typed  $Y$  that directly follows the first component. Therefore, we need to check that there is “something else” in the set of mentions, otherwise the sequence of tags could lead to an ambiguity in the encoding of continuous mentions. We ensure this via transition  $(8, \epsilon, 3)$ , that requires the generation of another component before reaching the final state. Finally, states 3, 4 and 9 recognizes extra  $X$  and  $Y$  in the set of mentions.

As such, the language of our grammar automaton is the set of well-formed tag sequences as described in Section 2.2. To use our grammar automaton, we need to remove  $\epsilon$ -transitions. The resulting WFSAs has 22 states.<sup>8</sup> In the case of structural component types, we can simply remove transition  $(1, DB-BY, 12)$  to constrain the leftmost mention to be labeled  $X$ .

**Practical implementation.** The intersection of

<sup>8</sup>Although 22 states is small and allows very fast computation, it is already too large for drawing a comprehensive figure.

the grammar and the sentence automata does not result in a homogeneous Markov chain as transition weights correspond to tag weights for the next word, and are therefore different at each step. However, the resulting automaton has always a simple time-invariant structure. In term of implementation, this reduces to applying a mask at each step, and both Viterbi and forward algorithms can be implemented using basic differentiable tensor operations. For MAP inference, we compute the path of maximum weight and then rely on backpropagation to retrieve the sequence of tags (Mensch and Blondel, 2018, Section 2.1).

## 4 Weakly-Supervised Learning

The negative log-likelihood (NLL) loss,

$$\ell(\mathbf{w}; \mathbf{y}) = -\langle \mathbf{y}, \mathbf{w} \rangle + A_Y(\mathbf{w}),$$

requires knowledge of the gold output  $\mathbf{y}$ . Unfortunately, NER datasets only contains annotated mentions, but not their component types (e.g. we do not know which components are body parts and events). Therefore, we need to resort on weakly-supervised learning to infer this information.

### 4.1 Learning with Partial Labels

Learning with partial labels refers to the case where the gold output is unknown but there is access to a subset of labels that includes the gold one (Grandvalet and Bengio, 2004; Nguyen and Caruana, 2008; Cour et al., 2011). Let  $\tilde{Y} \subseteq Y$  be the set of tag sequences that recovers the gold discontinuous mentions. For the example in Figure 1,  $\tilde{Y}$

contain two sequences, one where components of the set of mentions are labeled X/X/Y/Y and the other Y/Y/X/X. For a sentence containing  $k$  sets of mentions, we have  $|\tilde{Y}| = 2^k$ .

Following [Jin and Ghahramani \(2002\)](#), we minimize the NLL after marginalizing over  $\tilde{Y}$ :

$$\begin{aligned} \tilde{\ell}(\mathbf{w}; \tilde{Y}) &= -\log p_{\theta}(\tilde{Y}|\mathbf{x}) = -\log \sum_{\mathbf{y} \in \tilde{Y}} p_{\theta}(\mathbf{y}|\mathbf{x}) \\ &= A_{\tilde{Y}}(f_{\theta}(\mathbf{x})) - \log \underbrace{\sum_{\mathbf{y} \in \tilde{Y}} \exp\langle \mathbf{y}, f_{\theta}(\mathbf{x}) \rangle}_{=A_{\tilde{Y}}(f_{\theta}(\mathbf{x}))}, \end{aligned} \quad (3)$$

where  $A_{\tilde{Y}}$  is the *clamped* log-partition, which can be efficiently computed via a dynamic programming algorithm. In speech processing,  $A_{\tilde{Y}}$  is called the alignment model and the associated FSA the numerator graph ([Povey et al., 2016](#); [Hadian et al., 2018](#)).

**Relation with EM.** We can interpret minimizing  $\tilde{\ell}$  as an Expectation-Maximization (EM) procedure ([Neal and Hinton, 1998](#)). Indeed, the variational formulation of the clamped log-partition is:

$$A_{\tilde{Y}}(\mathbf{w}) = \sup_{\boldsymbol{\mu} \in \text{conv } \tilde{Y}} \langle \boldsymbol{\mu}, \mathbf{w} \rangle - \Omega_{\tilde{Y}}(\boldsymbol{\mu}),$$

where  $\text{conv}$  denotes the convex hull and  $\Omega_{\tilde{Y}}$  is a structured entropy term as described by [Blondel et al. \(2020, Section 7.1\)](#). Setting  $\mathbf{w} = f_{\theta}(\mathbf{x})$ , by Danskin’s theorem ([Danskin, 1966](#); [Bertsekas, 1999](#)), the gradient of the  $A$  is:

$$\hat{\boldsymbol{\mu}}_{\tilde{Y}}(\mathbf{w}) = \nabla A_{\tilde{Y}}(\mathbf{w}) = \arg \max_{\boldsymbol{\mu} \in \text{conv } \tilde{Y}} \langle \boldsymbol{\mu}, \mathbf{w} \rangle - \Omega_{\tilde{Y}}(\boldsymbol{\mu}).$$

We rewrite the minimization of  $\tilde{\ell}$  as a two-step procedure:

1. **E step:** compute  $\hat{\boldsymbol{\mu}}_{\tilde{Y}}(\mathbf{w})$ ;
2. **M step:** take one gradient step over the network parameters using the marginal distribution computed in E step, yielding the loss:

$$\ell(\mathbf{w}; \hat{\boldsymbol{\mu}}_{\tilde{Y}}(\mathbf{w})) = -\langle \mathbf{y}, \hat{\boldsymbol{\mu}}(\mathbf{w}) \rangle + A_{\tilde{Y}}(\mathbf{w}).$$

It is important to note that  $\hat{\boldsymbol{\mu}}_{\tilde{Y}}(\mathbf{w})$  is considered as a constant in the M step, *i.e.* the gradient is:

$$\nabla \ell(\mathbf{w}; \hat{\boldsymbol{\mu}}_{\tilde{Y}}(\mathbf{w})) = -\hat{\boldsymbol{\mu}}(\mathbf{w}) + \nabla A_{\tilde{Y}}(\mathbf{w}) = \nabla \tilde{\ell}(\mathbf{w}; \tilde{Y}),$$

meaning that this EM procedure is equivalent to minimizing the loss in Equation (3).

This suggests a “Hard EM” alternative, where the  $E$  step computes the unregularized maximum:

$$\hat{\mathbf{y}}_{\tilde{Y}}(\mathbf{w}) = \arg \max_{\mathbf{y} \in \text{conv } \tilde{Y}} \langle \mathbf{y}, \mathbf{w} \rangle,$$

and then apply one step of gradient descent using the loss  $\ell(\mathbf{w}; \hat{\mathbf{y}}_{\tilde{Y}}(\mathbf{w}))$  in the M step.

## 4.2 Silver Annotation of Components

In order to automatically annotate components, we collect names of body parts from the metathesaurus MRCONSO.RRF of the Unified Medical Language System (UMLS, version 2023ab).<sup>9</sup> We select English entries corresponding to semantic types “Body Location or Region”, “Body Part, Organ, or Organ Component” and “Body Space or Junction”, via the annotation in the lexicon MRSTY.RRF, which corresponds to identifiers T029, T023 and T030, respectively.<sup>10</sup> However, we remove all acronyms (indicated via the marker ABR) as they would introduce too many false positives in the annotation process (*e.g.* “in” and “am” are acronyms of body parts). This leads to 218 134 names of body parts.

Then, we try to match words of components with these entries. If at least one word of a component match an entry, we consider it as a body part. Note that a single match fully disambiguate a set of mentions.

## 5 Related Work

**Tagging methods.** [Tang et al. \(2013\)](#) proposed the BIOHD tagging scheme for discontinuous NER. A major issue of their approach is its *structural ambiguity*: several tag sequences can encode the same discontinuous mention, and different discontinuous mentions have the same associated tag sequence, see ([Muis and Lu, 2016, Section 3.1](#)). A choice to resolve ambiguity has to be made when making a prediction, meaning that there are structures that cannot be predicted. Moreover, this approach does not constrain the output tag sequence to be well-formed, *i.e.* it may not be possible to reconstruct mentions from a predicted tag sequence. The tagging scheme used by [Metke-Jimenez and Karimi \(2016\)](#) and [Dai et al. \(2017\)](#) has the same limitation. [Muis and Lu \(2016\)](#) proposed a graph-based method that ensures that predictions are well-formed, but their approach still exhibits structural ambiguity.

<sup>9</sup>[https://www.ncbi.nlm.nih.gov/books/NBK9685/table/ch03.T.concept\\_names\\_and\\_sources\\_file\\_mr/](https://www.ncbi.nlm.nih.gov/books/NBK9685/table/ch03.T.concept_names_and_sources_file_mr/)

<sup>10</sup><https://www.ncbi.nlm.nih.gov/books/NBK9685/table/ch03.Tf/>

**Other methods.** Wang and Lu (2019) rely on a two-step model that first predicts continuous spans (*i.e.* components) and then uses a separate classifier that combines them together. Dai et al. (2020) proposed a novel transition-based model. These two approaches are based on sequential predictions that are trained using gold intermediate outputs, which can lead to error propagation once a single mistake is made at test time. To resolve this problem, Wang et al. (2021) proposed a method that jointly predicts spans and their combination based on the maximal clique problem. A downside of these approaches is that they are more computationally costly (and therefore slower) than tagging methods.

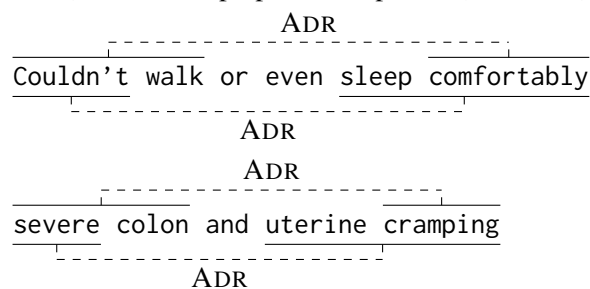
## 6 Experiments

We evaluate our approach on three standard English datasets for discontinuous named-entity recognition in the biomedical domain: CADEC (Karimi et al., 2015), SHARE2013 (Pradhan et al.) and SHARE2014 (Mowery et al.). We pre-process the data using the script of Dai et al. (2020). Note that our tagging scheme cannot predict all discontinuous mentions in the data, *i.e.* there are sentences that we cannot convert to our representation. Therefore, we remove these sentences from the training set.<sup>11</sup> Data statistics are given in Table 2.

### 6.1 Discontinuity Analysis

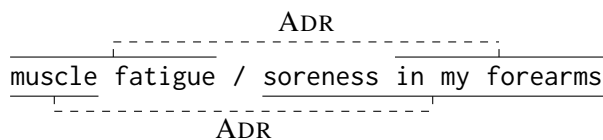
We conduct a qualitative analysis of the search space of our algorithm on the full CADEC dataset. There are 26 discontinuous NER structures incompatible with our approach.<sup>12</sup>

There are discontinuous mentions where there is a *partially* shared component. This is due to shared negation (1 case), shared adjective (5 cases) and shared prepositional phrase (PP, 1 case):



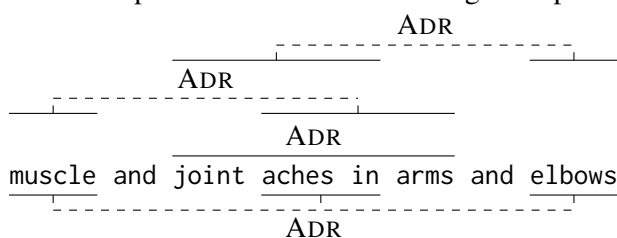
<sup>11</sup>Obviously, we do not remove anything from the test set.

<sup>12</sup>We do not count single mentions: we count full sets of mentions that cannot be recognized by our algorithm.



Although we cannot recognize these structures, we could extend our automaton to recognize the shared part as a continuous chunk (negation, adjective or PP), and the rest using our two layer representation.

There are also discontinuous mentions that are composed of three components (16 cases), which we cannot recognize. This can happen because there is a coordination in both subject and PP positions as in the following example:<sup>13</sup>



The mention “muscle aches in elbows” is composed of three components.

Finally, the last three incompatibilities are due to a convoluted syntactic structure and annotation errors (2 cases). Interestingly, some annotation errors can be detected thanks to our new annotation schema. For example, in CADEC the sequence “renal and respiratory failure” as been incorrectly annotated as containing renal respiratory failure instead of renal failure. In SHARE2014, the sequence “pleural / abdominal effusions” as been incorrectly annotated as containing effusions instead of abdominal effusions. Note that in this paper we used the datasets as such and did not fix any error so that results are comparable with previous work.

### 6.2 Results

Our neural network is excessively simple: we use the DEBERTA-V3 pretrained self-attentive network (He et al., 2021a,b) followed by a single linear projection that maps context-sensitive embeddings to tag weights. All training details are given in Appendix A. For each loss function, we train six models with six different seeds and we select the best model using the development set.

**Results.** We report the F-measure on all mentions and on discontinuous mentions only in Ta-

<sup>13</sup>This example has been slightly changed for formatting.



	CADEC		SHARE2013		SHARE2014	
	F1	Disc. F1	F1	Disc. F1	F1	Disc. F1
<b>Previous work</b>						
Tang et al. (2013)			75.0			
Tang et al. (2018)	66.3					
Metke-Jimenez and Karimi (2016)	64.4		56.5		60.2	
Metke-Jimenez and Karimi (2016) <sup>†</sup>	67.4	1.8	74.9	18.8	76.6	6.0
Muis and Lu (2016) <sup>†</sup>	58.0	23.9	70.3	50.0	74.7	41.1
Dai et al. (2020)	69.0	37.9	77.7	52.5	79.6	49.2
Wang et al. (2021)	71.5	44.4	81.2	55.9	81.3	54.1
<b>This work</b>						
Soft EM	71.1	38.1	80.7	49.2	81.5	51.9
Hard EM	71.9	35.9	82.0	51.9	81.6	54.1
Weakly soft EM	71.8	37.6	82.0	52.0	81.4	46.2
Weakly hard EM	70.4	33.6	82.0	52.1	81.8	49.8
Structural labels	72.9	41.5	82.1	53.3	80.9	53.7

Table 1: Results on on three different datasets. Results marked with <sup>†</sup> are reproductions by Wang et al. (2021).

Split	CADEC	SHARE2013	SHARE2014
Train	5340 (306)	8508 (477)	17407 (777)
- filtered	5322 (288)	8432 (401)	17294 (667)
Dev.	1097 (59)	1250 (58)	1361 (59)
Test	1160 (74)	9009 (301)	15850 (411)

Table 2: Number of sentences in each split. The number in parentheses corresponds to the number of sentences with at least one discontinuous mention.

Model	CADEC	S2013	S2014
Dai et al. (2020)	36	41	40
Wang et al. (2021)	193	200	198
This work	8286	10216	10206

Table 3: Speed comparison in terms of sentence per seconds. Numbers for Dai et al. (2020) are BERT-based models, as reproduced by Wang et al. (2021).

ble 1. The evaluation is conducted on the the original representation so results are comparable with previous work. Our approach leads to similar results to previous work. We do not observe significant differences between different loss functions.

**Speed.** All numbers are reported for computation on NVIDIA V100 GPUs. Training takes approximately 40, 60 and 80 minutes on CADEC, SHARE2013 and SHARE2014, respectively. Table 3 compares decoding with previous work of Dai et al. (2020) and Wang et al. (2021). The transition-based model of Dai et al. (2020) is particularly slow as their approach cannot fully exploit GPU parallelization. Our approach is  $\sim$ 40-50 times faster than the method of Wang et al. (2021). This is due to two reasons: (1) they use a complex neural net-

work architecture on top of a BERT-like model and (2) for each input they must solve a NP-hard problem (maximum clique) to make the prediction.

## 7 Conclusion

In this work, we propose a novel tagging scheme for discontinuous NER based on a two-layer representation of discontinuous mentions. Our approach leads to result on par with state-of-the-art using a very simple neural network architecture. Importantly, decoding with our model is very fast compared to previous work.

Our main objective with this work is to propose a simple plug-in method for discontinuous NER: any future work on models for BIO tagging can now also be trivially evaluated on discontinuous NER. Moreover, our approach is also fast to train, meaning that there is no significant cost overhead.

## Acknowledgments

I thank Vlad Niculae and François Yvon for their comments and suggestions. I thank Lucas Ondel-Yang for the many discussions on finite state-automata that inspired the decoding algorithm described in this paper. I thank Pierre Zweigenbaum for the help with Share datasets and the UMLS database.

Work partially done while I was a researcher at LISN and ISIR. This work was granted access to the HPC/AI resources of IDRIS under the allocation 2024-AD011013727R1 made by GENCI.

## Limitations

The approach proposed in this paper cannot cover all form of discontinuities observed in the three datasets. Indeed, some discontinuous mentions are composed of three parts or more. However, they are rare so our results are still competitive. Moreover, our contribution is focused on the general decoding approach that can be extended by future work.

Discontinuous NER datasets are scarce, therefore we are only able to experiment on three datasets in the biomedical domain in English. We suspect this is due to a *chicken or the egg* dilemma: discontinuity are often not annotated as there are no easy plug-and-easy approach to predict them, and there is little NLP work in the domain as there are only a few datasets available for experiments.

During the evaluation of our approach, we observed that many mentions are missing in the gold annotation. As such, all results reported on these datasets (including previous works) should be taken *with a pinch of salt*.

## References

- Arturo Argueta and David Chiang. 2017. [Decoding with finite-state transducers on GPUs](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1044–1052, Valencia, Spain. Association for Computational Linguistics.
- Leonard E Baum. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3(1):1–8.
- Jesse A. Berlin, Susan C. Glasser, and Susan S. Ellenberg. 2008. [Adverse event detection in drug development: Recommendations and obligations beyond phase 3](#). *American Journal of Public Health*, 98(8):1366–1371. PMID: 18556607.
- Dimitri P Bertsekas. 1999. *Nonlinear programming*. Athena Scientific Belmont.
- Mathieu Blondel, André F.T. Martins, and Vlad Niculae. 2020. [Learning with Fenchel-Young losses](#). *Journal of Machine Learning Research*, 21(35):1–69.
- N. Chinchor and P. Robinson. 1998. [Appendix E: MUC-7 named entity task definition \(version 3.5\)](#). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.
- Preciosa M Coloma, Gianluca Trifirò, Vaishali Patachia, and Miriam Sturkenboom. 2013. Postmarketing safety surveillance: where does signal detection using electronic healthcare records fit into the big picture? *Drug safety*, 36:183–197.
- Timothee Cour, Ben Sapp, and Ben Taskar. 2011. [Learning from partial labels](#). *Journal of Machine Learning Research*, 12(42):1501–1536.
- Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. [An effective transition-based model for discontinuous NER](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5860–5870, Online. Association for Computational Linguistics.
- Xiang Dai, Sarvnaz Karimi, and Cecile Paris. 2017. [Medication and adverse event extraction from noisy text](#). In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 79–87, Brisbane, Australia.
- John M. Danskin. 1966. [The theory of max-min, with applications](#). *SIAM Journal on Applied Mathematics*, 14(4):641–664.
- Samuel Eilenberg. 1974. *Automata, languages, and machines*. Academic press.
- Hao Fei, Donghong Ji, Bobo Li, Yijiang Liu, Yafeng Ren, and Fei Li. 2021. [Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12785–12793.
- G.D. Forney. 1973. [The Viterbi algorithm](#). *Proceedings of the IEEE*, 61(3):268–278.
- Xianping Ge. 2002. Segmental semi-markov models and applications to sequence analysis.
- Joshua Goodman. 1999. [Semiring parsing](#). *Computational Linguistics*, 25(4):573–606.
- Yves Grandvalet and Yoshua Bengio. 2004. Learning from partial labels with minimum entropy. Centre interuniversitaire de recherche en analyse des organisations (CIRANO).
- Ralph Grishman and Beth Sundheim. 1996. [Message Understanding Conference- 6: A brief history](#). In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. 2018. End-to-end speech recognition using lattice-free MMI. In *Interspeech*, pages 12–16.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). Preprint, arXiv:2111.09543.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. [Deberta: Decoding-enhanced bert with disentangled attention](#). In *International Conference on Learning Representations*.

- John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. 2001. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65.
- Jacques Janssen and Nikolaos Limnios. 1999. *Semi-Markov models and applications*. Kluwer Academic.
- Rong Jin and Zoubin Ghahramani. 2002. [Learning with multiple labels](#). In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.
- Stephan Kanthak and Hermann Ney. 2004. [FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 510–517, Barcelona, Spain.
- Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.
- Lauri Karttunen, Jean-Pierre Chanod, Gregory Grefenstette, and Anne Schille. 1996. [Regular expressions for language engineering](#). *Natural Language Engineering*, 2(4):305–328.
- Kimmo Koskenniemi. 1990. [Finite-state parsing and disambiguation](#). In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.
- Kimmo Koskenniemi, Pasi Tapanainen, and Atro Voutilainen. 1992. [Compiling and using finite-state syntactic rules](#). In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.
- George Lakoff and Stanley Peters. 1969. Phrasal conjunction and symmetric predicates. modern studies in english: Readings in transformational grammar.
- Winfried Lechner. 2000. Conjunction reduction in subordinate structures. In *North East Linguistics Society*, volume 30, page 5.
- Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. [Unified named entity recognition as word-relation classification](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10965–10973.
- Chu-Cheng Lin, Hao Zhu, Matthew R. Gormley, and Jason Eisner. 2019. [Neural finite-state transducers: Beyond rational relations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 272–283, Minneapolis, Minnesota. Association for Computational Linguistics.
- Matilde Marcolli and Ryan Thorngren. 2014. Thermodynamic semirings. *Journal of Noncommutative Geometry*, 8(2):337–392.
- Arthur Mensch and Mathieu Blondel. 2018. [Differentiable dynamic programming for structured prediction and attention](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3462–3471. PMLR.
- Alejandro Metke-Jimenez and Sarvnaz Karimi. 2016. Concept identification and normalisation for adverse drug event discovery in medical forums. In *BMDID@ISWC*.
- Mehryar Mohri. 2009. [Weighted Automata Algorithms](#), pages 213–254. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Mehryar Mohri, Fernando Pereira, and Michael Riley. 1996. [Weighted automata in text and speech processing](#). In *Proceedings of the ECAI 1996 Workshop*.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines](#). In *International Conference on Learning Representations*.
- Danielle L Mowery, Sumithra Velupillai, Brett R South, Lee Christensen, David Martinez, Liadh Kelly, Lorraine Goeriot, Noemie Elhadad, Sameer Pradhan, Guergana Savova, and Wendy W Chapman. Task 2: Share/clef ehealth evaluation lab 2014.
- Aldrian Obaja Muis and Wei Lu. 2016. [Learning to recognize discontinuous entities](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 75–84, Austin, Texas. Association for Computational Linguistics.
- Radford M. Neal and Geoffrey E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.
- Nam Nguyen and Rich Caruana. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–559.
- Sean Papay, Roman Klinger, and Sebastian Pado. 2022. [Constraining linear-chain CRFs to regular languages](#). In *International Conference on Learning Representations*.
- Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. 2016. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Interspeech*, pages 2751–2755.
- Sameer Pradhan, Noemie Elhadad, Brett R South, David Martinez, Lee M Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana K Savova. Task 1: ShARE/CLEF eHealth evaluation lab 2013.

- Lance Ramshaw and Mitch Marcus. 1995. [Text chunking using transformation-based learning](#). In *Third Workshop on Very Large Corpora*.
- Pushpendre Rastogi, Ryan Cotterell, and Jason Eisner. 2016. [Weighting finite-state transductions with neural context](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 623–633, San Diego, California. Association for Computational Linguistics.
- Lev Ratinov and Dan Roth. 2009. [Design challenges and misconceptions in named entity recognition](#). In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.
- Alexander Rush. 2020. [Torch-struct: Deep structured prediction library](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342, Online. Association for Computational Linguistics.
- Sunita Sarawagi and William W Cohen. 2004. [Semi-markov conditional random fields for information extraction](#). In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.
- Buzhou Tang, Jianglu Hu, Xiaolong Wang, and Qingcai Chen. 2018. Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF. *Wireless Communications & Mobile Computing (Online)*, 2018.
- Buzhou Tang, Yonghui Wu, Min Jiang, Joshua C Denny, and Hua Xu. 2013. Recognizing and encoding disorder concepts in clinical text using machine learning and vector space model. *CLEF (Working Notes)*, 665.
- Pasi Tapanainen. 1997. [Applying a Finite-State Intersection Grammar](#). In *Finite-State Language Processing*. The MIT Press.
- Roy Tromble and Jason Eisner. 2006. [A fast finite-state relaxation method for enforcing global constraints on sequence decoding](#). In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 423–430, New York City, USA. Association for Computational Linguistics.
- Martin J Wainwright, Michael I Jordan, et al. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.
- Bailin Wang and Wei Lu. 2019. [Combining spans into entities: A neural two-stage approach for recognizing discontinuous entities](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6216–6224, Hong Kong, China. Association for Computational Linguistics.
- Qing Wang, Haojie Jia, Wenfei Song, and Qi Li. 2023. [CoRec: An easy approach for coordination recognition](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15112–15120, Singapore. Association for Computational Linguistics.
- Yucheng Wang, Bowen Yu, Hongsong Zhu, Tingwen Liu, Nan Yu, and Limin Sun. 2021. [Discontinuous named entity recognition as maximal clique discovery](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 764–774, Online. Association for Computational Linguistics.
- Christopher Wilder. 2018. [Conjunction Reduction and Right-Node Raising](#). In *The Oxford Handbook of Ellipsis*. Oxford University Press.



## A Training details

The model is trained for 20 epochs using the cosine learning rate scheduler as implemented in the HuggingFace library. The maximum learning rate is fixed to  $10^{-5}$ . The warmup ratio is 10%. We apply dropout with a probability of 0.5 to BERT's output. The gradient norm is clipped to 1. All parameters have a weight decay of 0.01. We use the Adam variant proposed by [Mosbach et al. \(2021\)](#).