

Span-based Named Entity Recognition by Generating and Compressing Information

Nhung T. H. Nguyen¹, Makoto Miwa^{2,3} and Sophia Ananiadou^{1,3,4}

¹National Centre for Text Mining, Department of Computer Science,
The University of Manchester, United Kingdom

²Toyota Technological Institute, Japan

³Artificial Intelligence Research Center (AIRC),
National Institute of Advanced Industrial Science and Technology (AIST), Japan

⁴Alan Turing Institute, United Kingdom

{nhung.nguyen, sophia.ananiadou}@manchester.ac.uk
makoto-miwa@toyota-ti.ac.jp

Abstract

The information bottleneck (IB) principle has been proven effective in various NLP applications. The existing work, however, only used either generative or information compression models to improve the performance of the target task. In this paper, we propose to combine the two types of IB models into one system to enhance Named Entity Recognition (NER). For one type of IB model, we incorporate two unsupervised generative components, span reconstruction and synonym generation, into a span-based NER system. The span reconstruction ensures that the contextualised span representation keeps the span information, while the synonym generation makes synonyms have similar representations even in different contexts. For the other type of IB model, we add a supervised IB layer that performs information compression into the system to preserve useful features for NER in the resulting span representations. Experiments on five different corpora indicate that jointly training both generative and information compression models can enhance the performance of the baseline span-based NER system. Our source code is publicly available at <https://github.com/nguyenth/joint-ib-models>.

1 Introduction

Tishby et al. (1999) introduced the information bottleneck (IB) method to compress representation while preserving meaningful information. The method has been incorporated in many state-of-the-art (SOTA) deep models such as Variational Autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014), and Deep Variational Information Bottleneck (Deep VIB) (Alemi et al., 2017). Those deep models can be divided into supervised generative models (e.g., Deep VIB) and

unsupervised ones (e.g., VAEs) (Voloshynovskiy et al., 2019).

Both VAE and VIB have been applied to NLP applications. For example, Effland and Collins (2019) and Chen et al. (2018) proposed to use VAE in sequence labelling tasks such as POS tagging and NER. Meanwhile, Wang et al. (2022) used VIB to tackle OOV issues in NER. The two types of IB have also been employed in other tasks, such as dialogue response generation (Chen et al., 2022), parsing (Li and Eisner, 2019), paraphrase generation for MT (Ormazabal et al., 2022), and text summarisation (West et al., 2019), to name a few. Such previous work only used one type of IB model (either VAE or VIB) in their system, and it is unclear whether we can effectively combine the two types of IB models.

The NER task has been typically approached using sequence models such as BiLSTM (Lample et al., 2016) and BERT (Devlin et al., 2019; Lee et al., 2019; Beltagy et al., 2019). At the same time, we have seen the rise of span-based models (Sohrab and Miwa, 2018; Zheng et al., 2019; Tan et al., 2020; Xia et al., 2019; Xu et al., 2021; Fu et al., 2021; Li et al., 2021), which are simple and effective. Using a span-based model, we can directly represent and manipulate span representations.

This paper investigates the effects of combining the two IB types for the NER task. To that end, we jointly train the span-based NER system with two VAE components and one VIB component. The first VAE component is span reconstruction, used to reconstruct original spans. This component is similar to Sentence VAEs (Bowman et al., 2016), but the model learns only from spans instead of sentences. The second VAE one is syn-

onym generation, used to generate synonym(s) of a span. We first collect synonyms of each span from an external knowledge base (KB) and then train a VAE model that can generate the corresponding synonyms given a span. By adding the synonym generation into the model, we indirectly inject semantic information from synonyms into the span representation. The last component is a VIB component (Mahabadi et al., 2021) introduced into the system to compress span representations while keeping useful features for NER.

We evaluate the proposed model on five different corpora: BC5CDR (Li et al., 2016), GENIA (Kim et al., 2003), MedMention-21st (Mohan and Li, 2019), NCBI Disease (Dogan et al., 2014), and ShARe/CLEFE (Suominen et al., 2013). For the synonym generation component, we identify synonyms of each span by performing exact matching against mentions in Unified Medical Language System (UMLS) (Bodenreider, 2004). Experimental results show that by incorporating the two IB types into a span-based NER, we can improve the performance over the baseline span-based model on BC5CDR-Disease, GENIA, MedMention, and NCBI datasets. In the case of GENIA—one of the most popular corpora for nested entities, the proposed model could perform favourably compared with current SOTA systems, even with the simple span-based baseline model without any recent enhancements like boundary detection (Tan et al., 2020; Xu et al., 2021). Furthermore, such boundary-enhanced models can benefit from our approach.

We additionally performed some analysis on the intermediate output of the proposed model. We observed that when jointly trained VAEs with a NER task, the latent variable is restructured satisfactorily towards the task. Similarly, posteriors estimated by the VIB component are clustered neatly even though the input information is compressed. Such distinguishable clusters are potentially helpful for entity linking (Liu et al., 2021). We also found that synonym generation helped improve the quality of span reconstruction and the NER performance.

In summary, the contributions of our paper are:

- This is the first study that investigates the impact of combining two IB methods on NER.
- Through experiments on five different corpora, we demonstrate that the joint model can improve the baseline performance in most cases.
- In-depth analyses on the intermediate output

of the joint model indicate that each component plays a different role in enhancing span representations as expected.

2 Related Work

2.1 Span-based NER

Traditionally, sequence models such as BiLSTM (Lample et al., 2016) and BERT (Devlin et al., 2019; Lee et al., 2019; Beltagy et al., 2019) have been used to tackle the task of NER, producing state-of-the-art (SOTA) performance. However, those models could not perform on overlapping entities, i.e., a span has more than one named entity category or nested entities. To address this issue, Sohrab and Miwa (2018) proposed the span-based approach. In this approach, all possible spans are exhaustively generated given a specific span length. The span representation was calculated based on a pre-trained language model and then classified to a corresponding entity type by a linear layer. Following the suite, numerous studies have shown that span-based approaches to NER could produce SOTA performance (Zheng et al., 2019; Tan et al., 2020; Xia et al., 2019; Sohrab et al., 2019; Xu et al., 2021; Fu et al., 2021; Li et al., 2021; Yu et al., 2022).

Fu et al. (2021) designed SpanNER that learns the representation of a span based on its token representation and the span length embedding. SpanNER can also make ensemble predictions from both span-based and sequence labelling systems. Similarly, Yu et al. (2022) proposed SNER that represents a span by considering the context embedding from BERT, i.e., the CLS embedding. Ouchi et al. (2020) proposed to learn the similarity between spans using instance-based learning. The model will assign the class label based on its similar training span at inference time. Xu et al. (2021) used a supervised multi-head self-attention mechanism, where each head corresponds to one entity type, to construct the word-level correlations for each type.

Following the work mentioned above, this paper also focuses on span-based models. Our model, however, learns span representation differently. In particular, we update the span representation using span reconstruction and synonym generation. We then compress it by selecting relevant features using the IB method. It is noted that our model can potentially be incorporated into existing span-based models.

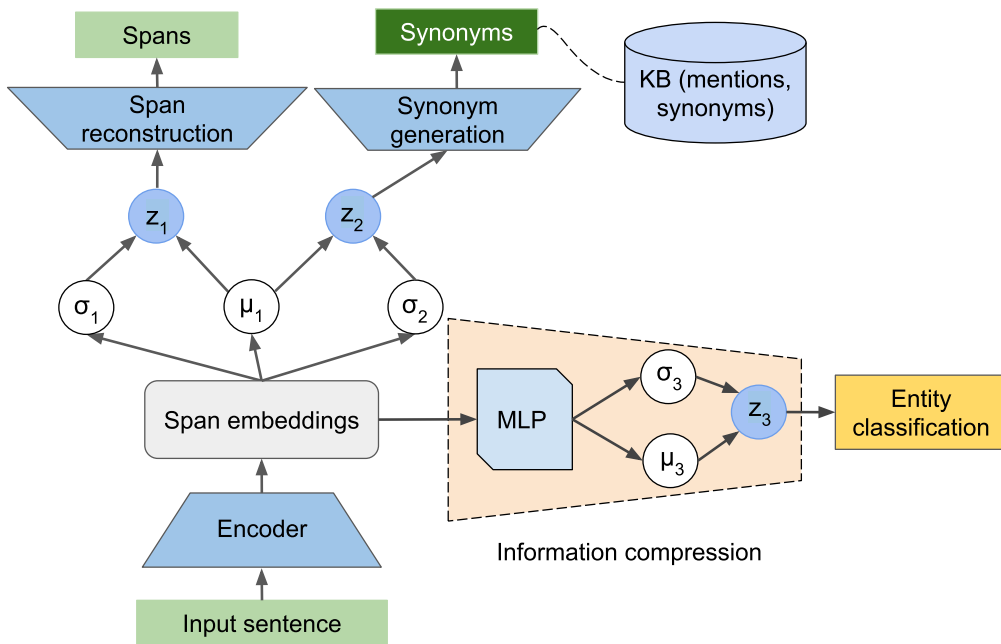


Figure 1: Span-based NER is jointly trained with one supervised VIB component: information compression, and two unsupervised VAE components: span reconstruction and synonym generation. The encoder is shared across the components. Synonyms are collected from an external knowledge base (KB).

2.2 Information Bottleneck in NLP

The Information Bottleneck (IB) method has been applied to various NLP applications. We divide those studies into two main groups: generative models and information compression models.

In the first group, we have studies that used IB as generative models, namely Variational Autoencoders (VAEs). [Effland and Collins \(2019\)](#) proposed a sequence labelling NER model that treats a neural CRF as the amortised approximate posterior in a discrete structured VAE. Meanwhile, [Chen et al. \(2018\)](#) applied neural variational methods to sequence labelling by combining a latent variable generative model and a discriminative labeller. VAEs were also employed in dialogue response generation ([Chen et al., 2022](#)) and relation extraction ([Yuan and Eldardiry, 2021](#); [Christopoulou et al., 2021](#)).

The second group are studies that used IB as information compression, namely Variational Information Bottleneck (VIB). [Mahabadi et al. \(2021\)](#) was one of the first studies using VIB to fine-tune low-resource target tasks. [Wang et al. \(2022\)](#) employed the information bottleneck principle to tackle OOV issues in NER. Information compression was also used in parsing ([Li and Eisner, 2019](#)), paraphrase generation for MT ([Ormazabal et al., 2022](#)), research replication prediction ([Luo et al.,](#)

[2022](#)), text classification ([Zhang et al., 2022](#)), improving the attention’s reliability ([Zhou et al., 2021](#)) and text summarisation ([West et al., 2019](#)).

Unlike previous studies, we combine generative and information compression models for the task of NER. In our model, span reconstruction ensures that the span representation keeps the span information, while synonym generation makes similar synonyms have similar representations even in different contexts. Meanwhile, information compression helps suppress irrelevant features, addressing overfitting (if any). To the best of our knowledge, this is the first paper investigating such a joint system for span-based NER.

3 Methods

The overall framework of our model is illustrated in Figure 1. We first encode span embeddings using a transformer-based network. The output from the encoder is then used in three components: (1) entity classification that classifies input spans into an entity type or non-entity using a supervised VIB framework ([Mahabadi et al., 2021](#)); (2) span reconstruction that recovers input spans of gold entities; and (3) synonym generation that generates synonyms for a gold entity. Similarly to a multi-task setting, we simultaneously train the three components. We only run the first component at the inference step to predict named entities.

3.1 Span Reconstruction and Synonym Generation

3.1.1 Encoder

Given an input sentence with n words: $\{w_0, w_1, \dots, w_n\}$, we use a multi-layer transformer-based encoder (Devlin et al., 2019) to encode the sentence. As a result, we have the following contextualised sub-word vectors $\{v_0, v_1, \dots, v_m\}$. Similarly to Sohrab and Miwa (2018), we then exhaustively extract all possible spans with a maximum length of sl from the input sentence. Each span embedding is calculated as follows:

$$s_{i,j} = \left[v_i; \frac{\sum_{t=i}^j v_t}{j-i+1}; v_j \right], \quad (1)$$

where i and j are the start and end positions of the span, v_t is the embedding of the t -th sub-word, and $[\cdot; \cdot]$ denotes the concatenation operation.

We then apply two linear layers on top of the span embeddings to construct the parameters of a posterior distribution $q(z|s)$ using the following equations:

$$\begin{aligned} \mu &= W_\mu s + b_\mu, \\ \sigma^2 &= W_\sigma s + b_\sigma, \end{aligned} \quad (2)$$

where μ and σ are the parameters of a multivariate Gaussian, representing the feature space of the span; W and b are weights and biases of the linear layers, respectively. The posterior distribution is approximated via a latent variable z using the reparameterisation trick (Kingma and Welling, 2014) as follows

$$z = \mu + \sigma\epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, 1). \quad (3)$$

In the case that we train both span reconstruction and synonym generation, we have different parameters of σ_1 and σ_2 , respectively. To encourage span and synonyms to distribute closely, μ is shared between them.

3.1.2 Decoders

Our decoder is an LSTM network that greedily reconstructs the input span or generates all the corresponding synonyms (if any) in an autoregressive manner. Given latent z from the encoder, we first use z to initialise the hidden state of the decoder via a linear layer transformation. We then form the input of the decoder with the teacher forcing strategy (Williams and Zipser, 1989), i.e., we concatenate z with the representation of each word w_t ¹

¹We pass a special start symbol as w_0 .

in a given gold span (for span reconstruction) or a gold synonym (for synonym generation).

3.1.3 Learning

To train VAEs, we maximise the Evidence Lower Bound (ELBO) that includes two losses: the reconstruction loss and the Kullback-Leibler divergence (D_{KL}). The reconstruction loss is the cross entropy loss between an actual span and its reconstruction or synonym. The D_{KL} is calculated based on a prior distribution ($p(z)$) and the posterior distribution ($q(z|s)$) produced by the decoder. In the case of span reconstruction (SR), the loss for a span s is

$$\begin{aligned} L_{SR}(\theta_1, \phi_1) &= \mathbb{E}_{z_1 \sim q_{\phi_1}(z_1|s)} [\log(p_{\theta_1}(s|z_1))] \\ &\quad - D_{KL}(q_{\phi_1}(z_1|s) \parallel p_{\theta_1}(z_1)). \end{aligned} \quad (4)$$

Similarly, with the synonym generation (SG), the loss is

$$\begin{aligned} L_{SG}(\theta_2, \phi_2) &= \mathbb{E}_{z_2 \sim q_{\phi_2}(z_2|s)} [\log(p_{\theta_2}(s|z_2))] \\ &\quad - D_{KL}(q_{\phi_2}(z_2|s) \parallel p_{\theta_2}(z_2)). \end{aligned} \quad (5)$$

In Equations 4 and 5, θ and ϕ are weights and biases of the network, respectively. We attach a subscript to each parameter to denote that the parameter belongs to the span reconstruction component or the synonym generation one.

3.2 Entity Classification with Supervised IB

The main objective of the supervised IB is to preserve the information about the target class(es) in the latent while filtering out irrelevant information from the input (Voloshynovskiy et al., 2019). As a result, the objective loss function for supervised IB is based on the compression loss and the prediction loss, as shown in Equation 6.

$$\begin{aligned} L_{VIB}(\theta_3, \phi_3) &= \beta \mathbb{E}_s [D_{KL}(p_{\theta_3}(z_3|s), r(z_3))] \\ &\quad + \mathbb{E}_{z_3 \sim p_{\theta_3}(z_3|s)} [-\log q_{\phi_3}(y|z_3)], \end{aligned} \quad (6)$$

where $r(z_3)$ is an estimate of the prior probability $p_{\theta_3}(z_3)$, β is in a range of $[0, 1]$, and y is the true label of the input span. Similarly to previous equations, θ_3 and ϕ_3 are weights and biases of the network, respectively.

Following Mahabadi et al. (2021), we use a multi-layer perceptron (MLP) with two linear layers to compute the compressed representation of a span. We approximate $q_{\phi_3}(y|z_3)$ using another

Corpus	Train			Dev			Test			Entity
	#Doc.	#Ent.	Syn_cov	#Doc.	#Ent.	Syn_cov	#Doc.	#Ent.	Syn_cov	
NCBI	592	5,134	79.25%	100	787	74.46%	100	960	75.62%	Disease
BC5-Dis	500	4,182	87.18%	500	4,244	87.18%	500	4,424	89.69%	Disease
BC5-Chem	500	5,203	93.14%	500	5,347	94.58%	500	5,385	94.00%	Chemical
ShARe	149	3,630	77.96%	50	1,413	77.35%	99	4,912	81.54%	Disorder
GENIA	1,599	45,036	55.98%	189	4,274	49.46%	212	5,346	51.05%	5 types
MM	2,635	122,241	70.21%	878	40,884	70.55%	879	40,157	71.17%	21 types

Table 1: Statistics numbers of experimental corpora. #Doc. indicates the number of documents; #Ent. indicates the number of gold entities; Syn_cov indicates the percentage of gold entities that have synonyms in UMLS.

linear layer. In particular, this layer is a binary classifier with a sigmoid function to predict the correct entity category of an input span. We use binary cross entropy (BCE) loss to compute the prediction loss (i.e., the second term of Equation 6) in our model.

3.3 Training Objective

We jointly train span reconstruction, synonym generation, and entity classification with the sum of the following optimisation objective for all the spans:

$$L = L_{VIB} + \gamma(L_{SR} + L_{SG}), \quad (7)$$

where γ is a hyper-parameter with a range of $[0, 1]$. It is noted that during training, L_{VIB} is calculated for all spans, while L_{SR} and L_{SG} are calculated for gold spans only.

4 Experiments

4.1 Datasets

We conducted experiments on five different datasets: NCBI Disease (Dogan et al., 2014), BC5CDR (Li et al., 2016), ShARe/CLEFE (ShARe) (Suominen et al., 2013), GENIA (Kim et al., 2003), and MedMention-21st (MM) (Mohan and Li, 2019). Following previous work, we divided BC5CDR into BC5CDR-Disease (BC5-Dis) and BC5CDR-Chemical (BC5-Chem). To detect synonyms of each span, we only performed exact matching against mentions from UMLS (Bodenreider, 2004). In Table 1, we report the number of documents, the number of golden entities, and the percentage of golden entities that have synonyms in the UMLS (version 2017AA) of the experimental corpora. Since UMLS does not cover every entity, there are cases that an entity does not have any synonyms².

²In this case, we only lowercase texts in the data and UMLS before matching. Using other techniques to find synonyms in UMLS is beyond the scope of this paper.

Corpus	Baseline	Joint model
BC5-Chemical	91.38	91.30
BC5-Disease	84.46	85.25
GENIA	76.93	77.43
MedMention-21st	62.78	63.21
NCBI	88.12	88.29
ShARe/CLEFE	80.88	80.77
Average	80.76	81.04

Table 2: F1 scores (%) on the test sets. Bold numbers indicate the joint model is better than the baseline.

4.2 Settings

Regarding the encoder, we employed the pre-trained SciBERT model (Beltagy et al., 2019). Regarding the decoder, we used an LSTM with one hidden layer; input vectors to the LSTM were extracted from resulting vectors trained on a combination of PubMed, PMC texts and the English Wikipedia (Pyysalo et al., 2013).

It is noted that before jointly training with the entity classification, we pre-trained the auxiliary components, i.e., span reconstruction and synonym generation in several epochs. The NER performance was measured based on the exact matching F1 scores calculated by the N2C2 Shared Task NER evaluation script (Henry et al., 2020). All hyperparameter settings are detailed in Appendix A.

4.3 Results

Across all corpora, we compare the joint model in Figure 1 with a Baseline system that is a span-based model³ using SciBERT, as reported in Table 2. With the GENIA corpus, we also collected the performance of other span-based SOTA systems for comparison, as shown in Table 3.

In Table 2, we can see that performance across the corpora differs. The proposed model produced better NER performance than the baseline ones

³We used the span-based NER implementation by Trieu et al. (2020)

System	F1 score
Instance-based (Ouchi et al., 2020)	74.20
Boundary-aware (Zheng et al., 2019)	74.70
BENSC (Tan et al., 2020)	78.30
MHSA (Xu et al., 2021)	79.60
Baseline	76.93
Joint model	77.43

Table 3: F1 scores (%) on the GENIA testing set compared with other SOTA systems.

on four corpora. The exception happened with BC5CDR-Chemical and ShARe/CLEFE where the simple span-based model obtained higher F1 scores than the joint model. The corpora’s characteristics are one possible reason for this. For BC5-Chemical, reconstructing chemical entities and finding their correct synonyms are more complex than the other entities (see Section 5.3). For ShAe/CLEFE, its documents are health records, not scientific papers like the others. On average, however, the joint model performed better than the baseline.

Results in Table 3 show that on the GENIA corpus, the joint model could perform better than (1) the baseline and (2) the two span-based models: the instance-based NER model (Ouchi et al., 2021) and the boundary-aware one (Zheng et al., 2019). However, our model produced lower F1 scores than BENSC (Tan et al., 2020) and MHSA (Xu et al., 2021). This can be explained by the fact that both BENSC and MHSA specifically address nested entities by enhancing span boundary detection, while our model classifies all possible spans. Nevertheless, it is noted that our objective is not to improve the SOTA; we focus on investigating the joint model for the NER task anyway. Furthermore, our model can be incorporated into those span-based SOTA systems.

5 Analysis

5.1 Ablation Study

To evaluate the effect of each component on the proposed model, we ran the following settings on the development sets:

- SupVIB: the span-based model using the supervised IB entity classification.
- SupVIB_0: β was set to 0, meaning that the compression loss was not involved in the training stage. This setting is similar to the baseline setting, but the span embeddings are compressed by the MLP before being fed into the

linear layer.

- SupVIB + SpanReco: we jointly trained span reconstruction with the SupVIB.
- SupVIB + SpanReco + SynGen (All): the full joint model with all three components.

From Table 4, we can observe the following. Firstly, using SupVIB, we could obtain higher F1 scores in most of the corpora than the baseline. The compression loss is important for SupVIB. In four over six corpora, setting $\beta = 0$ made the performance drop. Secondly, when we introduced the span reconstruction component alone into the model, except for MedMention-21st and NCBI, the component degraded the performance. We observed that SupVIB restructured embeddings differently than SupVIB+SpanReco, which probably affected the performance. Moreover, SupVIB was slightly better than SupVIB+SpanReco in distinguishing entities’ boundaries.

When we introduced synonym generation into the model (i.e., the All setting), we could obtain better performance than the baseline model on five over six corpora. This indicates that indirectly injecting semantic information into the span representation is mostly helpful. The exceptions were with MedMention and ShARe/CLEFE, where the All setting produced a lower F1 score than the SupVIB+SpanReco.

By jointly training two types of IB, i.e., generative model and supervised IB, we could enhance the NER performance over the baseline on four out of six testing corpora. Especially with the two most complex, multi-category NER corpora, i.e., GENIA and MedMention-21, the joint model could outperform the baseline by a larger margin than the other corpora.

Lastly, looking at the last column in Table 4, i.e., the average F1 score across the corpora, we can see that the scores were increased from the top to the bottom. We, therefore, can confirm that adding the proposed components one after another helped improve the performance in general.

5.1.1 Impact of the Shared Parameters in VAEs

To encourage spans and their synonyms to distribute closely, we have μ shared between the span reconstruction and the synonym generation, as in the aforementioned All model. Theoretically, we can also share σ between the two components. To investigate the impact of these shared parameters, we conducted two more experiments:

Setting	BC5-Chem	BC5-Dis	GENIA	MM	NCBI	ShARe	Avg.
Baseline	93.34	84.88	78.00	62.83	87.89	80.89	81.30
SupVIB_0	94.01	85.51	78.26	62.31	87.76	81.14	81.50
SupVIB	94.05	85.51	78.36	63.00	87.70	81.65	81.71
SupVIB + SpanReco	93.69	85.23	78.22	63.63	88.05	81.56	81.73
All	93.78	85.56	79.30	63.39	88.41	81.11	81.93
All (w/ shared σ)	93.87	85.33	78.90	63.21	87.62	80.93	81.64
All (w/o shared μ and σ)	93.49	85.03	78.97	61.85	87.89	81.78	81.50

Table 4: F1 scores (%) on the development sets produced by different settings. All is the Joint model in Tables 2 and 3. The last column is the average score across the corpora. Bold numbers indicate the best performance in each corpus.

- All with shared σ : the full model in which the two VAE components share μ and σ .
- All without shared μ and σ : the full model in which each VAE component has its own (i.e., independent) μ and σ .

From the last two rows in Table 4, we can see that the All model with shared σ made the F1 scores drop across the corpora (except for the BC5CDR-Chemical) compared with the All model. This is expected because sharing μ , i.e., sharing the mean of the two distributions, is more reasonable than sharing both of the parameters, i.e., sharing both the mean and the variance of the two distributions. However, when we have the independent μ and σ , there is no common pattern in the performance. Compared with the All model, the All without shared μ and σ model produced lower scores on most of the corpora, except for the ShARe/CLEFE one. On average, the shared μ setting (i.e., the All model) was superior to the other two.

5.2 NER Error Analysis

We classified false positive predictions by NER models into two classes:

- Category errors: denote predictions that have correct spans but wrong category,
- Span errors: denote predictions that have incorrect spans.

Regarding the BC5CDR, NCBI, and ShARe/CLEFE corpora, all false positives are due to span errors since these corpora have only one NER category. We report the percentage of span errors in false positives on the development sets of these corpora in Table 5. On BC5-Chemical and ShARe/CLEFE, the proposed model produced more span errors than the baseline one. This situation somehow aligns with the exceptionally lower performance of the proposed model on the

Corpus	Baseline	All
BC5-Chemical	6.48	6.50
BC5-Disease	15.21	13.52
NCBI	12.76	12.36
ShARe/CLEFE	18.02	19.23

Table 5: Percentage of span errors on the development sets. The lower the number, the better the model.

Corpus	Category Error		Span Error	
	Base	All	Base	All
GENIA	126	115	4,133	4,411
MM	2,447	2,028	37,148	34,747

Table 6: Number of category errors and span errors on the development sets. Base means the Baseline model. The lower the number, the better the model.

two corpora reported in Table 2.

In cases of MedMention-21st and GENIA, we report the number of category and span errors in Table 6. We can see that the proposed model produced fewer category errors than the baseline one on both corpora. We hypothesise that the synonym generation component attributed to this improvement; the semantic information provided by the component enhanced the ability to distinguish named entities’ categories of the All model. Moreover, we find that categories in GENIA are more ambiguous than those in MedMention-21st. For example, it is difficult to distinguish between “Cell line” and “Cell type” or between “Protein” and “DNA”. This can explain the lower reduction of category errors in GENIA than in MedMention-21st.

Table 6 also shows that the proposed model could help reduce span errors on MedMention-21st but not on GENIA. We think that the effectiveness of our model depends on the annotation schemes. As mentioned above, the GENIA cor-

Corpus	SupVIB+SpanReco	All
BC5-Chemical	0.0066	0.0051
BC5-Disease	0.0092	0.0243
GENIA	0.3902	0.4479
MedMention-21st	0.0612	0.1482
NCBI	0.1244	0.2228
ShARe/CLEFE	0.0033	0.1283

Table 7: BLEU-2 scores of span reconstruction on the development set of each corpus.

pus contains gene/protein-related entities, of which the boundary of an entity is more specially defined than those in MedMention-21st. As a result, detecting correct spans in GENIA is more challenging than in MedMention. For example, the All model detected “HBxAg - specific synthetic polypeptides” as a Protein entity while the correct span should be “HBxAg” only.

Nevertheless, the span errors accounted for a majority of the false positives across all corpora, indicating that an enhanced-boundary approach can help alleviate the situation.

5.3 Generation Quality

To evaluate the generation quality of the VAEs component, we calculated BLEU-2 scores of the reconstructed gold entities on the development set of each corpus. From Table 7, we can see that using synonym generation helped improve the BLEU scores in almost corpora. This indicates that semantic information from the synonym generation component is useful for the NER task and the span reconstruction.

In Table 8, we show examples where having synonym generation, i.e., the All model, could reconstruct original entities much better than SupVIB+SpanReco. In all examples, the reconstructed spans by the All model are more meaningful and fluent than those by SupVIB+SpanReco.

Among the six corpora, reconstructing chemical entities is the most challenging task, especially with short-form chemicals such as PPA (Phenylpropanolamine). When looking for synonyms of these entities in UMLS, we only did exact matching without checking any semantics. Therefore, there are cases in which we found UMLS synonyms for these abbreviations, but their full forms are completely different. For example, some UMLS synonyms of PPA are “primary progressive apha-

sia”, “primary progressive apraxia of speech”, and “Mesulam syndrome”, which are not relevant to “Phenylpropanolamine”—the correct full form of the chemical PPA. Nevertheless, we observed that the All model usually reconstructed an abbreviation if the original entity is an abbreviation, while the SupVIB+SpanReco did it very randomly. For example, in row 9 in Table 8, given an abbreviation of “PPA”, the All model could generate the exact full form given span, while SupVIB+SpanReco generated “antidepressant”—an incorrect full form for “PPA”.

5.4 Entity Posteriors

We further explore the effect of the proposed model by inspecting its intermediate values, which are posteriors z generated by the VAE and VIB components.

5.4.1 Span Reconstruction and Synonym Generation

We plot the resulting posteriors z_1 of gold entities on the development set in a 2D space using tSNE (van der Maaten and Hinton, 2008) in two cases: (i) Only VAEs: when we only train span reconstruction and synonym generation, and (ii) our joint model. We observed that across the settings, thanks to the target task of NER, z_1 was satisfactorily restructured, i.e., posteriors of entities in the same category are clustered together. This phenomenon is more visible in the case of GENIA and MedMention-21⁴ than in the other corpora since the two corpora have more than one named entity categories, as illustrated in the first row of Figure 2. In Figures 2a and 2c, entities are scattered over the space. Meanwhile, in Figures 2b and 2d, points in the same category are grouped. We can conclude that after simultaneously training VAEs and NER (the All model), we have better clusters than only training VAEs (Only VAEs).

5.4.2 Supervised IB

Similarly, we plot the gold entity embeddings and their posteriors z_3 estimated by the supervised IB in the second row of Figure 2. We can see in Figures 2e and 2h that initially, the gold entity embeddings are not clustered very well. In contrast, their posteriors are neatly grouped as illustrated in Figures 2f and 2h. These clusters look even more distinguishable than those by z_1 , i.e., Figures 2b and

⁴We only show embeddings of the top-10 named entity categories in MedMention-21.

No.	Original span	SupVIB + SpanReco	All
1	nervous fibers	central channel	nervous muscles
2	renal failure	pain failure	renal failure
3	immunomodulatory therapy	Cg therapy	antiviral therapy
4	central hydrophobic core	central its carbon	central temporal core
5	pulmonary veins	liver ganglion	pulmonary vein
6	fibre degeneration	hearing loss	macular degeneration
7	cardiac inter - beat	cardiac - time coil	cardiac - - range
8	volatile compounds	translational compounds	volatile compounds
9	PPA	antidepressant	PPA
10	autoimmune lymphoproliferative syndrome	ankylosing lymphoproliferative syndrome	autoimmune lymphoproliferative syndrome

Table 8: Some examples to show that the synonym information is helpful to span reconstruction.

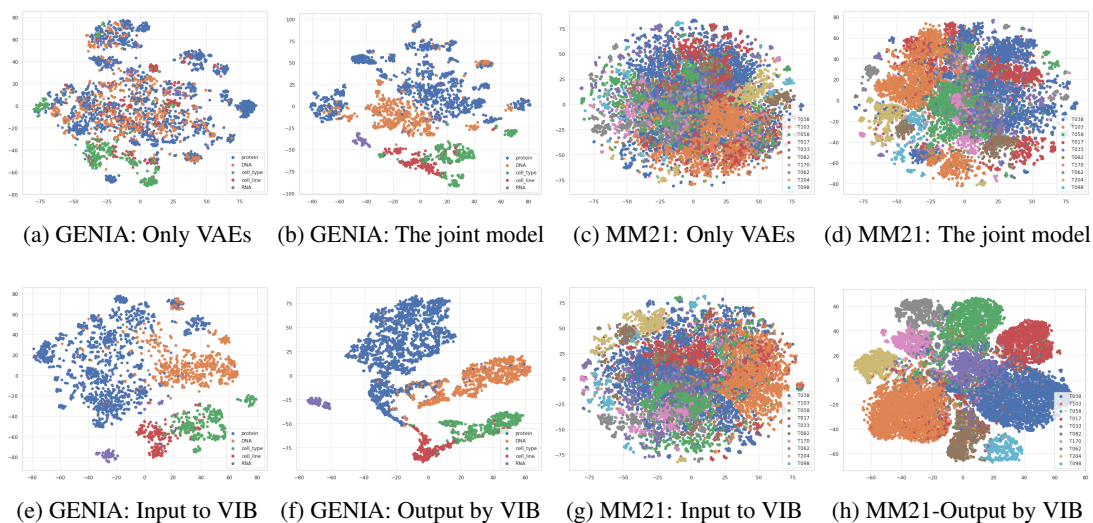


Figure 2: Visualisation of entity posteriors on the development set of GENIA and MedMention-21. Points in the same colour indicate entities in the same category. First row: posteriors from the generation models. Second row: the embeddings of entities input to VIB and the corresponding posteriors by VIB.

2d. This phenomenon is understandable because z_3 is approximated based on supervised learning while z_1 is based on unsupervised one.

These visualisations also demonstrate the effectiveness of the supervised IB method. While reducing the input representation size, the method can still reserve relevant features about the target classes. As a result, we can have smaller but more meaningful representations than the original ones. Such neat clusters are potentially helpful for entity linking (Liu et al., 2021).

6 Conclusion

We introduced a joint span-based NER model consisting of three components: VAE-based span reconstruction, VAE-based synonym generation, and VIB-based NER. Each component plays a differ-

ent role in learning span representation. The VIB-based NER tries to preserve the information about the target NER category in the estimated latent while filtering out irrelevant information from the input. The model is forced to keep contextualised span information when having the span reconstruction component. Meanwhile, the synonym generation component indirectly injects semantic information about a span’s synonyms. When testing on five different corpora, we found that the joint model could perform better than the baseline.

The proposed model focuses on learning span representation, which is applicable not only to span-based NER but also to other span-based tasks such as event coreference resolution (Lu and Ng, 2021) and question answering (Li and Choi, 2020). We plan to apply our model to such tasks in the future.

Limitations

We find two main limitations of the proposed model. Firstly, we need synonyms for each gold entity to train the synonym generation component. Unfortunately, this requirement is unsuitable for many NER corpora, such as CoNLL 2012 and ACE2005. One possible way to go around this is to treat the coreferences of an entity (if any) as synonyms. Secondly, similarly to some span-based NER models, our model suffers from a considerable number of all possible spans. We can alleviate this limitation by detecting spans' boundary (Tan et al., 2020; Xu et al., 2021) before classifying spans. We, however, leave those as future work.

Acknowledgements

We would like to thank Dr. Hai-Long Trieu for his help and fruitful comments on the initial implementation of the paper. This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. [Optuna: A next-generation hyperparameter optimization framework](#). In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.
- Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. [Deep variational information bottleneck](#). In *Proceedings of the International Conference on Learning Representations*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Olivier Bodenreider. 2004. [The Unified Medical Language System \(UMLS\): integrating biomedical terminology](#). *Nucleic Acids Res.*, 32(Database-Issue):267–270.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. [Generating sentences from a continuous space](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.
- Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. [Variational sequential labelers for semi-supervised learning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 215–226, Brussels, Belgium. Association for Computational Linguistics.
- Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. 2022. [DialogVED: A pre-trained latent variable encoder-decoder model for dialog response generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4852–4864, Dublin, Ireland. Association for Computational Linguistics.
- Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2021. [Distantly supervised relation extraction with sentence reconstruction and knowledge base priors](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 11–26, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. [NCBI Disease Corpus: A Resource for Disease Name Recognition and Normalization](#). *Journal of biomedical informatics*, 47:1–10.
- Thomas Efland and Michael Collins. 2019. [Semi-Supervised Named Entity Recognition with CRF-VAEs](#). In *OpenReview*.
- Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. [SpanNER: Named entity re-/recognition as span prediction](#). 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 7183–7195, Online. Association for Computational Linguistics.
- Sam Henry, Kevin Buchan, Michele Filannino, Amber Stubbs, and Özlem Uzuner. 2020. [2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records](#). *Journal of the American Medical Informatics Association*, 27(1):3–12.
- Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. [GENIA corpus - a semantically annotated corpus for bio-textmining](#). In *Proceedings of*

- the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia*, pages 180–182.
- Diederik P. Kingma and Max Welling. 2014. [Auto-Encoding Variational Bayes](#). In *The 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [BioBERT: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*, 36(4):1234–1240.
- Changmao Li and Jinho D. Choi. 2020. [Transformers to learn hierarchical contexts in multiparty dialogue for span-based question answering](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5709–5714, Online. Association for Computational Linguistics.
- Fei Li, ZhiChao Lin, Meishan Zhang, and Donghong Ji. 2021. [A span-based model for joint overlapped and discontinuous named entity recognition](#). 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 4814–4828, Online. Association for Computational Linguistics.
- Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wieggers, and Zhiyong Lu. 2016. [BioCreative V CDR task corpus: a resource for chemical disease relation extraction](#). *Database*, 2016.
- Xiang Lisa Li and Jason Eisner. 2019. [Specializing word embeddings \(for parsing\) by information bottleneck](#). 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 2744–2754, Hong Kong, China. Association for Computational Linguistics.
- Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. [Self-alignment pretraining for biomedical entity representations](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.
- Jing Lu and Vincent Ng. 2021. [Span-based event coreference resolution](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13489–13497.
- Tianyi Luo, Rui Meng, Xin Wang, and Yang Liu. 2022. [Interpretable research replication prediction via variational contextual consistency sentence masking](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3864–3876, Dublin, Ireland. Association for Computational Linguistics.
- Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2021. [Variational information bottleneck for effective low-resource fine-tuning](#). In *Proceedings of the International Conference on Learning Representations*.
- Sunil Mohan and Donghui Li. 2019. [MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts](#). In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019*.
- Aitor Ormazabal, Mikel Artetxe, Aitor Soroa, Gorka Labaka, and Eneko Agirre. 2022. [Principled paraphrase generation with parallel corpora](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1638, Dublin, Ireland. Association for Computational Linguistics.
- Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. [Instance-based learning of span representations: A case study through named entity recognition](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6459, Online. Association for Computational Linguistics.
- Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Masashi Yoshikawa, and Kentaro Inui. 2021. [Instance-based neural dependency parsing](#). *Transactions of the Association for Computational Linguistics*, 9:1493–1507.
- Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. [Distributional semantics resources for biomedical text processing](#). In *Proceedings of the LBM 2013*.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. [Stochastic backpropagation and approximate inference in deep generative models](#). In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Beijing, China. PMLR.
- Mohammad Golam Sohrab and Makoto Miwa. 2018. [Deep exhaustive model for nested named entity recognition](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

- Mohammad Golam Sohrab, Minh Thang Pham, Makoto Miwa, and Hiroya Takamura. 2019. [A neural pipeline approach for the PharmaCoNER shared task using contextual exhaustive models](#). In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.
- Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J. F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeriot, David Martinez, and Guido Zuccon. 2013. [Overview of the ShARe/CLEF eHealth Evaluation Lab 2013](#). In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 212–231, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. [Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9016–9023.
- Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. [The information bottleneck method](#). In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377.
- H-L. Trieu, T. T. Tran, K. N. A. Duong, A. Nguyen, M. Miwa, and S. Ananiadou. 2020. [Deepeventmine: End-to-end neural nested event extraction from biomedical texts](#). *Bioinformatics*, page btaa540.
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-SNE](#). *Journal of Machine Learning Research*, 9:2579–2605.
- Slava Voloshynovskiy, Mouad Kondah, Shideh Rezaei-far, Olga Taran, Taras Holotyak, and Danilo Jimenez Rezende. 2019. [Information bottleneck through variational glasses](#). In *Proceedings of the 4th workshop on Bayesian Deep Learning (NeurIPS 2019)*.
- Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuanjing Huang. 2022. [MINER: Improving out-of-vocabulary named entity recognition from an information theoretic perspective](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5590–5600, Dublin, Ireland. Association for Computational Linguistics.
- Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. [BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle](#). 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 3752–3761, Hong Kong, China. Association for Computational Linguistics.
- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.
- Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and Philip Yu. 2019. [Multi-grained named entity recognition](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1430–1440, Florence, Italy. Association for Computational Linguistics.
- Yongxiu Xu, Heyan Huang, Chong Feng, and Yue Hu. 2021. [A Supervised Multi-Head Self-Attention Network for Nested Named Entity Recognition](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14185–14193.
- Jie Yu, Bin Ji, Shasha Li, Jun Ma, Huijun Liu, and Hao Xu. 2022. [S-ner: A concise and efficient span-based model for named entity recognition](#). *Sensors*, 22(8).
- Chenhan Yuan and Hoda Eldardiry. 2021. [Unsupervised relation extraction: A variational autoencoder approach](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1929–1938, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. [Improving the adversarial robustness of NLP models by information bottleneck](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598, Dublin, Ireland. Association for Computational Linguistics.
- Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. [A boundary-aware neural model for nested named entity recognition](#). 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 357–366, Hong Kong, China. Association for Computational Linguistics.
- Jie Zhou, Yuanbin Wu, Qin Chen, Xuanjing Huang, and Liang He. 2021. [Attending via both fine-tuning and compressing](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2152–2161, Online. Association for Computational Linguistics.

A Hyper-parameter settings

In Table 9, we show the hyper-parameter values used in our experiments. In both SupVIB+SpanReco and All settings, we trained VAEs in 10 epochs before introducing NER into the models. Our models were finetuned on NVIDIA Tesla V100 GPUs with 16GB of RAM using OpTuna (Akiba et al., 2019). Depending on the size

of the corpus, we ran different numbers of trials, which were in the range of [5,15].

Hyper-parameter name	Value
β	[1e-6, 1e-4]
γ	[1e-6, 1e-4]
Latent size	[512, 768, 1024]
Input size of the LSTM encoder	200
Output size of the LSTM decoder	256
NER learning rate	[1e-5, 3e-4]
VAE learning rate	[1e-4, 1e-3]
Batch size	16
Maximum length of a span	14
Maximum sentence length	512
Number of epochs	20
Number of epochs to pretrain VAEs	10

Table 9: Hyper-parameter values