Learning Informative Representations of Biomedical Relations with Latent Variable Models

Harshil Shah
University College London
BenevolentAI[*]
h.shah@cs.ucl.ac.uk

Julien Fauqueur
BenevolentAI
julien@benevolent.ai

Abstract

Extracting biomedical relations from large corpora of scientific documents is a challenging natural language processing task. Existing approaches usually focus on identifying a relation either in a single sentence (mention-level) or across an entire corpus (pair-level). In both cases, recent methods have achieved strong results by learning a point estimate to represent the relation; this is then used as the input to a relation classifier. However, the relation expressed in text between a pair of biomedical entities is often more complex than can be captured by a point estimate. To address this issue, we propose a latent variable model with an arbitrarily flexible distribution to represent the relation between an entity pair. Additionally, our model provides a unified architecture for both mention-level and pair-level relation extraction. We demonstrate that our model achieves results competitive with strong baselines for both tasks while having fewer parameters and being significantly faster to train. We make our code publicly available.

1 Introduction

The vast amounts of scientific literature can provide a significant source of information for biomedical research. Using this literature to identify relations between entities is an important task in various applications (van Mulligen et al., 2012; Segura-Bedmar et al., 2013; Bravo et al., 2015; Krallinger et al., 2017).

Existing approaches to biomedical relation extraction usually fall into one of two categories. Mention-level extraction aims to classify the relation between a pair of entities within a short span of text (usually a sentence). In contrast, pair-level extraction aims to classify the relation between a pair of entities across an entire paragraph, document or corpus.

For both mention-level and pair-level relation extraction, recent work has been focused on representation learning. This is considered to be one of the major steps towards making progress in artificial intelligence (Bengio et al., 2013). Representations of relations which understand their context are particularly important in biomedical research, where identifying fruitful targets is crucial due to the high costs of experimentation. Learning such representations is likely to require large amounts of unsupervised data due to the scarcity of labelled data in this domain.

Recent mention-level methods have been based on using large unsupervised models with Transformer networks (Vaswani et al., 2017) to learn representations of sentences containing pairs of entities. These representations are then used as the inputs to much smaller models, which perform supervised relation classification (Lee et al., 2019; Beltagy et al., 2019).

Recent pair-level methods have been based on encoding each mention of a pair of entities, and designing a mechanism to pool these encodings (across a paragraph, document, or corpus) into a single representation. This representation is then used to classify the relation between the entity pair (Verga et al., 2018; Jia et al., 2019).

However, representation learning methods for both mention-level and pair-level extraction typically use a point estimate for each representation. As a result, they may struggle to capture the nature of the true, potentially complex relations between each pair of entities. For example, Figure 1 shows sentences for two entity pairs which demonstrate that relation statements can be very different, typically depending on biological circumstances (*e.g.* anatomical location, experimental details, presence of a disease, *etc*). Such nuanced relations can be difficult to capture with a single point estimate.

We hypothesise that there is a true underlying

[*] Work completed during internship at BenevolentAI.

Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, pages 19–28
Online, November 20, 2020. ©2020 Association for Computational Linguistics

Protein **Akt** and protein **GSK3β**:

"… **Akt** negatively regulates **GSK3β** activity…"
"… **Akt** phosphorylates **GSK3β**…"

Protein **EAAT2** and disease **ALS**:

"**EAAT2**/C1-4 were found to be equally expressed in **ALS** patients and controls."
"**EAAT2** protein is significantly reduced in **ALS** in the motor cortex and spinal cord."

Figure 1: Two sets of sentences demonstrating the potentially complex nature of the relation between a pair of entities.

relation for each entity pair, and that this relation can be multimodal (because of the aforementioned complexities). The sentences containing each pair are textual observations of these underlying relations.

We therefore propose a probabilistic model which uses a continuous latent variable to represent the true relation between each entity pair. The distribution of a sentence containing that pair is then conditioned on this latent variable. In order to be able to model the complex relations between each entity pair, we use an infinite mixture distribution for the latent representation.

Our model provides a unified architecture for learning representations of relations between entity pairs both at mention and pair level. We show that (an approximation to) the posterior distribution of the latent variable can be used for mention-level relation classification. We also demonstrate that the prior distribution from the same model can be used for pair-level classification. On both tasks, we achieve results competitive with strong baselines with a model which has fewer parameters and is significantly faster to train.

The code is released at `https://github.com/BenevolentAI/RELVM`

2 Model

In this section, we introduce our unified architecture for both mention-level and pair-level relation extraction. Throughout, we use the following notation:

- c represents a 'context', *i.e.* a sentence (or sequence of tokens) containing a pair of entities. c has tokens $c_1, \ldots, c_T$.
  - c_{t_x} and c_{t_y} are the tokens representing the two entities. We replace the actual tokens denoting the two entities with generic `<ENT>` tokens. Therefore, a context is given by:

$$c = c_1, \ldots, c_{t_x-1}, \texttt{<ENT>}, c_{t_x+1}, \ldots, c_{t_y-1}, \texttt{<ENT>}, c_{t_y+1}, \ldots, c_T$$

- x and y are the input representations of the two entities.
  - For pair-level classification, x and y will be unique identifiers for the two entities.
  - For mention-level classification, x and y will be the types of the two entities, *e.g.* `GENE` and `DISEASE`. This is done in order to allow for fair comparisons with previous methods, which use the entity types for mention-level classification (see Section 4.2 for further details).
  - x and y always refer to the first and second entities in c respectively.
- $\mathbf{e}(c_t)$ is the embedding of token c_t. $\mathbf{e}(x)$ and $\mathbf{e}(y)$ are the embeddings of the entities x and y.
- r represents the relation label.

Approach Large corpora of labelled relation statements are often scarce, whereas unlabelled sentences are usually plentiful. In order to leverage these unlabelled sentences, we first train an unsupervised model to learn representations of entity pairs and the contexts in which they occur. We then train much smaller models to classify relations using the representations from the unsupervised model.

2.1 Representation learning model

When training the unsupervised representation learning model, we assume access to a corpus of sentences in which entities have been tagged but there are no relation labels. We train the representation model to maximise the conditional log-likelihood $\log p(c|x, y)$. θ will refer to the set of parameters of the representation model which we wish to optimise. A graph of the representation model is shown in Figure 2 and a more detailed explanation is given below.

There are many ways to express the same relation between a given pair of entities. For example, the sentences "*John is Mary's brother*" and "*Mary*

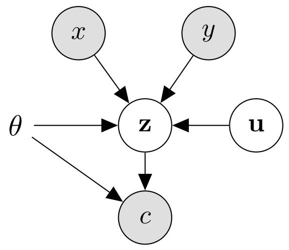

Figure 2: A graph depicting our unsupervised representation learning model. Clear nodes denote latent variables and shaded nodes denote observed variables. Representations from this model are used for both mention-level and pair-level relation classification.

is John's sister" express the same relation in different ways. In order to capture this phenomenon, we introduce a latent variable, $\mathbf{z}$, to represent the true underlying relation. This will be the representation used for mention-level and pair-level relation classification. The conditional distribution is parametrised as:

$$p(c|x,y) = \int_{\mathbf{z}} p_\theta(\mathbf{z}|x,y) p_\theta(c|\mathbf{z}) \tag{1}$$

Intuitively, $p_\theta(\mathbf{z}|x,y)$ captures the true underlying relation between the two entities x and y, and $p_\theta(c|\mathbf{z})$ captures the variation in the multiple possible ways of expressing that relation.

For computational simplicity, we could choose $p_\theta(\mathbf{z}|x,y)$ to be Gaussian. However in reality, the true relation between a pair of entities is probably more complex than can be modelled well with a unimodal distribution. We therefore introduce another latent variable $\mathbf{u}$ such that:

$$p_\theta(\mathbf{z}|x,y) = \int_{\mathbf{u}} p(\mathbf{u}) p_\theta(\mathbf{z}|x,y,\mathbf{u}) \tag{2}$$

For $p(\mathbf{u})$, we use a standard Gaussian distribution, $\mathcal{N}(\mathbf{0},\mathbf{I})$. For $p_\theta(\mathbf{z}|x,y,\mathbf{u})$, we again use a Gaussian distribution whose mean and variance are a function of x, y and $\mathbf{u}$. We concatenate together $\mathbf{e}(x)$, $\mathbf{e}(y)$, $\mathbf{e}(x) \odot \mathbf{e}(y)$ and $\mathbf{u}$, and pass the resulting vector into a feedforward network to output the mean and variance of $p_\theta(\mathbf{z}|x,y,\mathbf{u})$ ($\odot$ denotes element-wise multiplication). Using a nonlinear network allows the marginal distribution $p_\theta(\mathbf{z}|x,y)$ to be an infinite mixture distribution (Mattei and Frellsen, 2018). The objective becomes:

$$\log p(c|x,y) = \log \int_{\mathbf{u},\mathbf{z}} p(\mathbf{u}) p_\theta(\mathbf{z}|x,y,\mathbf{u}) p_\theta(c|\mathbf{z}) \tag{3}$$

We parametrise $p_\theta(c|\mathbf{z})$ with an LSTM, due to its strong performance in language modelling (Graves, 2013; Bowman et al., 2016; Melis et al., 2018). The conditional probabilities for $t = 1,\ldots,T$ are:

$$p_\theta(c_t = v|c_{1:t-1},\mathbf{z}) \propto \exp((\mathbf{W}\mathbf{h}_t^p) \cdot \mathbf{e}(v)) \tag{4}$$

where $\mathbf{W}$ is a learnable parameter of the model, and $\mathbf{h}_t^p$ is computed as:

$$\mathbf{h}_t^p = \text{LSTM}(\mathbf{z}, \mathbf{h}_{t-1}^p, \mathbf{e}(c_{t-1})) \tag{5}$$

Complete hyperparameter details are provided in Section 4.1.

2.1.1 Training

Because of the nonlinear functions involved in $p_\theta(\mathbf{z}|x,y,\mathbf{u})$ and $p_\theta(c|\mathbf{z})$, the integral in Equation (3) is intractable. We therefore perform approximate maximum likelihood estimation using stochastic gradient variational Bayes (SGVB) (Kingma and Welling, 2014; Rezende et al., 2014).

To do this, we parametrise a Gaussian inference distribution $q_\phi(\mathbf{u}|x,y,c)$ (referred to as $q_\phi(\mathbf{u})$ henceforth, for brevity) with trainable parameters ϕ. This allows us to maximise the following lower bound on the log-likelihood:

$$\begin{aligned}\log p(c|x,y) &\geq \mathbb{E}_{q_\phi(\mathbf{u})p_\theta(\mathbf{z}|x,y,\mathbf{u})}\left[\log \frac{p(\mathbf{u})p_\theta(c|\mathbf{z})}{q_\phi(\mathbf{u})}\right] \\ &\equiv \mathcal{L}_{\theta,\phi}(c,x,y) \end{aligned} \tag{6}$$

This bound can be approximated using Monte Carlo integration. It is optimised with respect to θ and ϕ jointly.

To parametrise $q_\phi(\mathbf{u})$, we use a bidirectional LSTM to encode the context. This is due to its ability to capture useful sentence level information into a low-dimensional vector (Zhou et al., 2016; Peters et al., 2018). It is computed as:

$$\overrightarrow{\mathbf{h}_t^q} = \text{LSTM}(\mathbf{e}(c_t), \overrightarrow{h_{t-1}^q}) \tag{7}$$

$$\overleftarrow{\mathbf{h}_t^q} = \text{LSTM}(\mathbf{e}(c_t), \overleftarrow{h_{t+1}^q}) \tag{8}$$

$$\mathbf{h}^q = [\overrightarrow{\mathbf{h}_T^q}; \overleftarrow{\mathbf{h}_1^q}] \tag{9}$$

We concatenate $\mathbf{h}^q$ to $\mathbf{e}(x)$, $\mathbf{e}(y)$ and $\mathbf{e}(x) \odot \mathbf{e}(y)$ and pass the resulting vector into a feedforward network to output the mean and variance of $q_\phi(\mathbf{u})$.

2.2 Mention-level classification

In this section, we assume that the unsupervised representation model from Section 2.1 has been trained with x and y being the types of the two entities. The representations $\mathbf{z}$ can now be used as the inputs to a supervised mention-level relation classification model.

For mention-level classification, we assume access to a corpus of sentences in which entities have been tagged and there are labels classifying the type of relation between the entity pair in each sentence. We train the mention-level classification model to maximise $p(r|x, y, c)$. λ will refer to the set of parameters of the mention-level classification model which we wish to optimise.

The representation $\mathbf{z}$ of the entity pair and context would ideally be distributed according to the posterior $p(\mathbf{z}|x, y, c)$ from the representation model. We would then optimise the parameters λ using the following objective:

$$p(r|x, y, c) = \int_{\mathbf{z}} p(\mathbf{z}|x, y, c) p_{\lambda}(r|\mathbf{z}) \tag{10}$$

However:

$$\begin{aligned} p(\mathbf{z}|x, y, c) &= \frac{p_{\theta}(\mathbf{z}, c|x, y)}{p(c|x, y)} && (11) \\ &= \frac{p_{\theta}(\mathbf{z}, c|x, y)}{\int_{\mathbf{u},\mathbf{z}} p(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u}) p_{\theta}(c|\mathbf{z})} && (12) \end{aligned}$$

As mentioned in Section 2.1.1, the integral in the denominator is intractable. Instead, the following approximation to the posterior can be used:

$$p(\mathbf{z}|x, y, c) \simeq \int_{\mathbf{u}} q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u}) \tag{13}$$

This is an approximation to the posterior because maximising the objective in Equation (6) is equivalent to minimising the KL divergence from $q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u})$ to $p(\mathbf{z}, \mathbf{u}|x, y, c)$ (Kingma and Welling, 2014):

$$\begin{aligned} &\mathcal{L}_{\theta,\phi}(c, x, y) = \log p(c|x, y) - \\ &\quad D_{\mathrm{KL}}[q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u}) || p(\mathbf{z}, \mathbf{u}|x, y, c)] && (14) \end{aligned}$$

Using this approximation, the mention-level classification objective becomes:

$$\begin{aligned} p(r|x, y, c) &\simeq \int_{\mathbf{u},\mathbf{z}} q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u}) p_{\lambda}(r|\mathbf{z}) && (15) \\ &= \mathbb{E}_{q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x,y,\mathbf{u})}[p_{\lambda}(r|\mathbf{z})] && (16) \end{aligned}$$

Empirically, however, we find that the model trains much more easily using the following objective:

$$\mathbb{E}_{q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x,y,\mathbf{u})}[\log p_{\lambda}(r|\mathbf{z})] \equiv \mathcal{L}_{\lambda}(r, c, x, y) \tag{17}$$

This is due, particularly at the start of training, to the values of $p_{\lambda}(r|\mathbf{z})$ being very small. Note that, due to Jensen's inequality, the objective in Equation (17) is in fact a lower bound on the log of the objective in Equation (16):

$$\mathcal{L}_{\lambda}(r, c, x, y) \leq \log \mathbb{E}_{q_{\phi}(\mathbf{u}) p_{\theta}(\mathbf{z}|x,y,\mathbf{u})}[p_{\lambda}(r|\mathbf{z})] \tag{18}$$

To parametrise $p_{\lambda}(r|\mathbf{z})$, we use a shallow feedforward network with a softmax function at the output. Complete hyperparameter details are provided in Section 4.2.

2.3 Pair-level classification

In this section, we assume that the unsupervised representation model from Section 2.1 has been trained with x and y being unique identifiers for the two entities. The representations $\mathbf{z}$ can now be used as the inputs to a supervised pair-level relation classification model.

For pair-level classification, we assume access to a dataset with pairs of entity identifiers, and labels classifying the type of relation between each pair. Instead of learning $p(r|x, y, c)$ as in mention-level classification, we now learn $p(r|x, y)$.

Intuitively, for pair-level classification, we wish to classify the relation between a pair of entities based on everything that the unsupervised model has learned about those entities (through the sentences containing them). This is unlike mention-level classification, where we classify the relation described in a specific sentence.

For pair-level classification, we follow a very similar approach to that described in Section 2.2 for mention-level classification. However we no longer base the input representation on the posterior distribution from the unsupervised model, $p(\mathbf{z}|x, y, c)$. Instead, the representation used will be distributed according to:

$$p_{\theta}(\mathbf{z}|x, y) = \int_{\mathbf{u}} p(\mathbf{u}) p_{\theta}(\mathbf{z}|x, y, \mathbf{u}) \tag{19}$$

Intuitively, this is the natural distribution to use, because we are interested in the relation between the entities x and y, without a specific context to condition on.

We denote ψ as the parameters of the pair-level supervised model. Then, following the same reasoning as Section 2.2, the objective for the pair-level supervised model is:

$$\mathbb{E}_{p(\mathbf{u}) p_{\theta}(\mathbf{z}|x,y,\mathbf{u})}[\log p_{\psi}(r|\mathbf{z})] \equiv \mathcal{L}_{\psi}(r, x, y) \tag{20}$$

To parametrise $p_\psi(r|\mathbf{z})$, we use a shallow feedforward network with a softmax function at the output. Complete hyperparameter details are provided in Section 4.3.

3 Related work

Mention-level relation extraction is typically performed using supervised learning. In the general domain, Zhang et al. (2017) combine an LSTM with a position-aware attention mechanism to perform multiclass relation extraction. Soares et al. (2019) fine-tune the BERT (Devlin et al., 2019) architecture to relation extraction tasks by enforcing similarity between representations of sentences containing the same pair of entities across a corpus. (Zhang et al., 2020) construct a teacher model to generate soft labels which guide the optimisation of a student network via knowledge distillation. In the biomedical and scientific domains, BioBERT (Lee et al., 2019) and SciBERT (Beltagy et al., 2019) train the BERT architecture on domain-specific corpora, achieving state of the art results on mention-level relation extraction tasks. Zhang et al. (2018) combine an RNN over the sentence's words and a CNN over its dependency graph to classify drug-drug and protein-protein interactions.

Pair-level relation extraction usually relies on distant supervision (Mintz et al., 2009). In the general domain, Hoffmann et al. (2011) develop a latent variable model to perform multi-instance learning while handling overlapping relations. Lin et al. (2016) use an attention mechanism to pool the representations of sentences containing a given pair into a single representation, which is then used as the input to a classifier. Quirk and Poon (2017) capture relations across sentences by linking dependency graphs between sentences. Other pair-level methods build representations using unsupervised models. Camacho-Collados et al. (2019) use a latent variable model to learn a point-estimate representation from the unigram distribution of tokens co-occurring in sentences with the given pair. Joshi et al. (2019) learn representations of pairs of entities by maximising their pointwise mutual information (PMI) with the contexts that the entities appear in. In the biomedical domain, Verga et al. (2018) build a paragraph-level representation using a modified Transformer network, and aggregate over mentions using a softmax function. Liang et al. (2019) combine knowledge embeddings and graph embeddings using a cascade learning framework to predict links in biochemical networks. Percha and Altman (2015) use a distributional semantics approach to cluster together drug-gene pairs which are related in similar ways.

Contrary to our work, there does not appear to be prior research performing both mention-level and pair-level relation extraction with a unified model.

4 Experiments

4.1 Representation learning model

We train the unsupervised representation model described in Section 2.1 using sentences from PubMed abstracts, PubMed Central (PMC) open-access full-text articles, and licensed full-text articles from Wiley and Springer. We take sentences with a maximum length of 140 tokens and tag the entities with their type using a dictionary-based method. Entities are linked to unique identifiers by first disambiguating entity types using a bidirectional LSTM sentence classifier, followed by type-specific term lookups. Note that if a sentence contains three or more entities, it is repeated in order to account for each possible pair of entities.

4.1.1 Architectures and training

To parametrise $p_\theta(\mathbf{z}|x, y, \mathbf{u})$, we use a 2-layer feedforward network with the ReLU nonlinearity. To parametrise $p_\theta(c|\mathbf{z})$, we use a 1-layer LSTM. To parametrise $q_\phi(\mathbf{u})$, we use a 1-layer bidirectional LSTM, the output of which is passed to a 2-layer feedforward network with the ReLU nonlinearity.

In order to evaluate the effect of the number of parameters on performance, we train four different versions of our representation learning model: {X-SMALL, SMALL, MEDIUM, LARGE}. These correspond to respective hidden state sizes of {128, 256, 512, 1024} in the networks. For all of the models, both $\mathbf{u}$ and $\mathbf{z}$, as well as all embeddings, are 300-dimensional.

We train the unsupervised representation models using a single sample approximation of the objective in Equation (6). We train for 400,000 iterations, using a minibatch size of 192 and optimising the parameters using Adam (Kingma and Ba, 2015) with a learning rate of 0.0001.

4.1.2 Optimisation challenges

The unsupervised objective in Equation (6) can be expressed as:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(c, x, y) = \mathbb{E}_{q_\phi(\mathbf{u})p_\theta(\mathbf{z}|x,y,\mathbf{u})}[\log p_\theta(c|\mathbf{z})] \\ - D_{\mathrm{KL}}[q_\phi(\mathbf{u})||p(\mathbf{u})]\end{aligned} \tag{21}$$

When training latent variable models with autoregressive observation distributions (such as that in Equation (4)), this objective can induce local optima where $q_\phi(\mathbf{u}) = p(\mathbf{u})$. This results in the KL divergence term in Equation (21) collapsing to 0, meaning the model ignores the latent variable altogether. To avoid such local optima, we use the following two methods (Bowman et al., 2016):

KL annealing We multiply the KL divergence term by a constant weight which is linearly annealed from 0 to 1 over the first 10,000 iterations of training. This helps the model to escape local optima where $D_{\mathrm{KL}}[q_\phi(\mathbf{u})||p(\mathbf{u})] = 0$ early in training.

Token dropout In Equation (5), we randomly drop the token embedding being passed to the next LSTM hidden state. We use a dropout rate of 50%. This encourages the LSTM to rely more on the representation $\mathbf{z}$ than the previous tokens when modelling the context.

4.1.3 Computational costs

We show the computational costs of our unsupervised representation models in Table 1. We compare against BioBERT (Lee et al., 2019), a language model with state-of-the-art performance on relation extraction.

All versions of our model have significantly fewer parameters than BioBERT. In terms of 'GPU days'[1], training BioBERT is approximately 25 to 40 times slower than training our model. In addition, inference is an order of magnitude faster with our model compared to BioBERT.

4.2 Mention-level classification

After training the unsupervised representation model (using the entity types for x and y), we use it to perform supervised mention-level relation classification, as described in Section 2.2. We use the EU-ADR (van Mulligen et al., 2012) and GAD (Bravo et al., 2015) datasets. In both datasets, each sentence contains a gene and disease. The task is to classify whether the given sentence either does or does not exhibit a relation between the gene and the disease. Examples from both datasets are shown in Table 2 and dataset statistics are shown in Table 3. As per previous work, we report the performance using 10-fold cross validation on each dataset (Lee et al., 2019).

We compare our results with those of BioBERT as reported by Lee et al. (2019). For a fair comparison, we use the same classifier architecture. This is a single layer network with a softmax nonlinearity. As well as training the parameters λ of the classifier, we also fine tune the parameters θ and ϕ of the representation model. Again, this is done to allow for a fair comparison with BioBERT (which follows the same procedure).

We approximate the objective in Equation (17) using 4 samples during training. We use a minibatch size of 8 and update the parameters using Adam with a learning rate of 0.00001. We train on EU-ADR for 200 iterations and on GAD for 3,000 iterations.

Note that the representations for BioBERT are 768-dimensional. This is in contrast to ours which are 300-dimensional.

4.2.1 Results

We perform 10-fold cross validation, and report the mean precision, recall and F1-score in Table 4. On EU-ADR, all versions of our model outperform BioBERT, with our LARGE model achieving a significantly higher F1-score. On this task, all versions of our model have significantly higher recall than BioBERT, with the precision being similar. On GAD, BioBERT slightly outperforms our LARGE model, thanks to its higher precision. In addition, we find that, on both tasks, the performance monotonically increases with the size of the unsupervised representation model.

These results show that it is possible to achieve results competitive with the state-of-the-art while making significant efficiency gains, both in terms of memory and time.

4.3 Pair-level classification

In this section, we use the LARGE representation model from Section 4.1, trained using the unique entity identifiers for x and y. We fix the parameters of the unsupervised representation model and use it to perform supervised pair-level classification, as described in Section 2.3.

We construct a multiclass classification dataset by combining multiple third-party biomedical datasets. These datasets only provide pairs of entities which are related. Therefore, if an entity pair does not appear in any of the datasets, they are assumed to be unrelated and given the label `NO-RELATION`. If two entities are related, the label is given by the concatenation of the two

[1] GPU days = No. of GPUs × training time (in days).

MODEL	PARAMS	TRAINING HARDWARE	TRAINING TIME	INFERENCE HARDWARE	INFERENCE TIME
BioBERT	110M	8 x V100 GPUs	10 days	1 x V100 GPU	0.0087s/sent.
Ours (X-SMALL)	2M	1 x V100 GPU	2 days	1 x V100 GPU	0.0004s/sent.
Ours (SMALL)	4M	1 x V100 GPU	2 days	1 x V100 GPU	0.0004s/sent.
Ours (MEDIUM)	10M	1 x V100 GPU	3 days	1 x V100 GPU	0.0005s/sent.
Ours (LARGE)	30M	1 x V100 GPU	3 days	1 x V100 GPU	0.0007s/sent.

Table 1: The computational costs of each of the unsupervised representation models we train. The inference time for each model is computed on a V100 GPU.

DATASET	x	y	c	r
EU-ADR	GENE	DISEASE	Based on <ENT> analyses, 41 <ENT> patients and 12 healthy controls were studied.	0
	DISEASE	GENE	<ENT> is associated with decreased expression of mucosal <ENT> .	1
GAD	GENE	DISEASE	A broad protective effect of <ENT> S180L against <ENT> per se is not discernible.	0
	GENE	DISEASE	The <ENT> polymorphism Tyr402His appears indicative of <ENT> pathogenesis.	1

Table 2: Positive ($r = 1$) and negative ($r = 0$) examples from the EU-ADR and GAD datasets.

DATASET	EU-ADR	GAD
# relations	355	5330

Table 3: Number of relations for the EU-ADR and GAD datasets.

entity types. This is therefore a multiclass classification problem, with the set of possible classes being {NO-RELATION, DISEASE-GENE, GENE-GENE, CHEMICAL-GENE, CHEMICAL-DISEASE}. Note that we only include entity pairs that occur in at least one sentence in the dataset used to train the representation learning model.

We randomly split the related entity pairs into training, validation and test sets. The set of entity pairs with label NO-RELATION is extremely large. We randomly assign a proportion of these to the validation and test sets. During training, we randomly sample a proportion of each minibatch from the remaining unrelated entity pairs. The dataset statistics are shown in Table 5.

For the pair-level classifier, we train a 2-layer model which has a 300-dimensional hidden layer with a skip connection. We approximate the objective in Equation (20) using 4 samples during training. We train for 100,000 iterations, using a minibatch size of 512 (of which 448 are sampled from the NO-RELATION set). We optimise the parameters using Adam with a learning rate of 0.0001. When making predictions on unseen data points, we only predict a label other than NO-RELATION if the predicted probability is higher than a threshold. This threshold is tuned to maximise the F1-score on the validation set.

4.3.1 Baselines

We compare our method with the following two baselines:

Co-occurrences For every entity pair that occurs in at least one sentence in the dataset used to train the representation learning model, we predict the relation to be positive (*i.e.* the concatenation of the types of the two entities). By design, this method will have perfect recall.

Attention This method is similar to those presented by Lin et al. (2016) and Verga et al. (2018). For a given pair of entities, we collect every sentence containing the pair from the dataset used to train the representation learning model. Each sentence is passed to an LSTM whose final state is taken as the sentence representation. The representations for all sentences for the given entity pair are pooled together into a single representation us-

Model	EU-ADR			GAD		
	P	R	F	P	R	F
BioBERT	80.92	90.81	84.83	**75.95**	88.08	**81.52**
Ours (X-SMALL)	79.62	98.08	87.71	67.83	90.76	77.45
Ours (SMALL)	80.35	98.09	88.14	68.31	91.75	78.16
Ours (MEDIUM)	80.72	98.46	88.59	69.68	91.82	78.72
Ours (LARGE)	**82.34**	**98.85**	**89.67**	72.26	**92.00**	80.79

Table 4: Results using 10-fold cross validation on the EU-ADR and GAD classification tasks. We report the mean precision (P), recall (R) and F1-score (F) over the 10 folds. For all metrics, higher is better.

Dataset	Pair-level
Train (excl. NO-RELATION)	263,112
Validation	691,627
Test	692,534

Table 5: Total counts across all relation types for the pair-level classification dataset. The training set excludes NO-RELATION types, as these are sampled during training.

Model	P	R	F
Co-occurrences	3.10	100.00	6.02
Attention	11.06	26.97	15.69
Ours	12.54	25.91	16.90

Table 6: Results on the test set of the pair-level classification task. We report the precision (P), recall (R) and the F1-score (F). For all metrics, higher is better.

ing an attention mechanism. This representation is then used as the input to a feedforward network with a softmax function at the output. This method is therefore trained on exactly the same dataset as our pair-level classifier.

The attention model is trained for 1,000,000 iterations using a minibatch size of 100 (of which 50 are sampled from the NO-RELATION set). The parameters are optimised using Adam with a learning rate of 0.000005. As with our model, when making predictions on unseen data points, we only predict a label other than NO-RELATION if the predicted probability is higher than a threshold. This threshold is tuned to maximise the F1-score on the validation set.

4.3.2 Results

The precision, recall, and F1-score on the test set are reported in Table 6. Our model achieves a higher F1-score than the attention model. Unsurprisingly, both the attention model and our model achieve significantly higher precision than the co-occurrence baseline at the expense of lower recall.

In contrast to the attention model, when classifying a new pair, our model does not need to encode all of the sentences containing that pair. This provides significant computational advantages, both in terms of memory and time.

5 Conclusion

We have presented a model for learning representations of pairs of biomedical entities from unlabelled text corpora. We use a latent variable with an arbitrarily flexible distribution in order to be able to capture the complex relations between each pair of entities. The unified architecture can be used for both mention-level and pair-level relation extraction. On both tasks, we achieve results competitive with strong baselines. We also show significant computational gains in terms of the number of parameters and training times.

Our model presents many avenues for future work. The results in Table 4 show that the model's performance improves with the size of the hidden states in the networks; this suggests that there are further gains achievable simply by providing the model with more parameters. The model could be further scaled up by using a hierarchy of latent variables to increase the expressive power of the representations.

Other directions include evaluating the benefits of having a representation which explicitly captures uncertainty about the relations. For example, this can be done by assessing if the model is less confident when making predictions about entity pairs which do not occur frequently in the unlabelled corpus. Additionally, since our model can produce a representation for any pair of entities (even those which do not occur together in the unlabelled corpus), it could be used in a link prediction setting to score unseen entity pairs.

Acknowledgements

We would like to thank our colleagues Sia Togia and Angus Brayne for their thorough feedback on this paper and Rogier Hintzen for his precious help in preparing the datasets.

References

I. Beltagy, K. Lo, and A. Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Y. Bengio, A. Courville, and P. Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35.

S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

À. Bravo, J. Piñero, N. Queralt-Rosinach, M. Rautschka, and L. Furlong. 2015. Extraction of Relations Between Genes and Diseases from Text and Large-Scale Data Analysis: Implications for Translational Research. *BMC Bioinformatics*, 16.

J. Camacho-Collados, L. Espinosa-Anke, S. Jameel, and S. Schockaert. 2019. A Latent Variable Model for Learning Distributional Relation Vectors. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

J. Devlin, M. Chang, K. Lee, and K. 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*.

A. Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR*, abs/1308.0850.

R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. Weld. 2011. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

R. Jia, C. Wong, and H. Poon. 2019. Document-Level N-ary Relation Extraction with Multiscale Representation Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

M. Joshi, E. Choi, O. Levy, D. Weld, and L. Zettlemoyer. 2019. pair2vec: Compositional Word-Pair Embeddings for Cross-Sentence Inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

D. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

D. Kingma and M. Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

M. Krallinger, O. Rabal, S. Akhondi, M. Pérez, J. Santamaría, G. Rodríguez, G. Tsatsaronis, A. Intxaurrondo, J. López, U. Nandal, E. van Buel, A. Chandrasekhar, M. Rodenburg, A. Lægreid, M. Doornenbal, J. Oyarzábal, A. Lourenço, and A. Valencia. 2017. Overview of the BioCreative VI chemical-protein interaction Track. In *Proceedings of the BioCreative VI Workshop*.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. So, and J. Kang. 2019. BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics*.

X. Liang, D. Li, M. Song, A. Madden, Y. Ding, and Y. Bu. 2019. Predicting Biomedical Relationships Using the Knowledge and Graph Embedding Cascade Model. *PLOS ONE*, 14.

Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

P. Mattei and J. Frellsen. 2018. Leveraging the Exact Likelihood of Deep Latent Variable Models. In *Advances in Neural Information Processing Systems 31*.

G. Melis, C. Dyer, and P. Blunsom. 2018. On the State of the Art of Evaluation in Neural Language Models. In *International Conference on Learning Representations*.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant Supervision for Relation Extraction Without Labeled Data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

E. van Mulligen, A. Fourrier-Reglat, D. Gurwitz, M. Molokhia, A. Nieto, G. Trifiro, J. Kors, and L. Furlong. 2012. The EU-ADR Corpus. *Journal of Biomedical Informatics*, 45.

B. Percha and R. Altman. 2015. Learning the Structure of Biomedical Relationships from Unstructured Text. *PLoS Computational Biology*, 11.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*.

C. Quirk and H. Poon. 2017. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

D. Rezende, S. Mohamed, and D. Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*.

I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo. 2013. SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation*.

L. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*.

P. Verga, E. Strubell, and A. McCallum. 2018. Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction. In *North American Chapter of the Association for Computational Linguistics*.

Y. Zhang, H. Lin, Z. Yang, J. Wang, S. Zhang, Y. Sun, and L. Yang. 2018. A Hybrid Model Based on Neural Networks for Biomedical Relation Extraction. *Journal of Biomedical Informatics*, 81.

Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu, Jiapeng Zhao, Quangang Li, and Li Guo. 2020. Distilling Knowledge from Well-Informed Soft Labels for Neural Relation Extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. 2016. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. In *Proceedings of the 26th International Conference on Computational Linguistics*.