REflex: Flexible Framework for Relation Extraction in Multiple Domains

Geeticka Chauhan
MIT CSAIL
geeticka@mit.edu

Matthew B. A. McDermott
MIT CSAIL
mmd@mit.edu

Peter Szolovits
MIT CSAIL
psz@mit.edu

Abstract

Systematic comparison of methods for relation extraction (RE) is difficult because many experiments in the field are not described precisely enough to be completely reproducible and many papers fail to report ablation studies that would highlight the relative contributions of their various combined techniques. In this work, we build a unifying framework for RE, applying this on three highly used datasets (from the general, biomedical and clinical domains) with the ability to be extendable to new datasets. By performing a systematic exploration of modeling, pre-processing and training methodologies, we find that choices of pre-processing are a large contributor performance and that omission of such information can further hinder fair comparison. Other insights from our exploration allow us to provide recommendations for future research in this area.

1 Introduction

Relation Extraction (RE) has gained a lot of interest from the community with the introduction of the Semeval tasks from 2007 by (Girju et al., 2007) and 2010 by (Hendrickx et al., 2009). The task is a subset of information extraction (IE) with the goal of finding semantic relationships between concepts in a given sentence, and is an important component of Natural Language Understanding (NLU). Applications include automatic knowledge base creation, question answering, as well as analysis of unstructured text data. Since the introduction of RE tasks in the general and medical domains, many researchers have explored the performance of different neural network architectures on the datasets (Socher et al., 2012; Zeng et al., 2014; Liu et al., 2016b; Sahu et al., 2016).

However, progress in RE is hampered by reproducibility issues as well as the difficulty in assessing which techniques in the literature will generalize to novel tasks, datasets and contexts. To assess the extent of these problems, we performed a manual review of 53 relevant neural RE papers[1] citing the three datasets (Hendrickx et al., 2009; Segura-Bedmar et al., 2013; Uzuner et al., 2011). The procedure for finding these papers is highlighted in (Chauhan, 2019).

Reproducibility Reproducibility is important for validating previous work and building upon it (Fokkens et al., 2013). Lack of reproducibility can be attributed to many factors such as difficulty in availability of source code (Ince et al., 2012) and omission of sources of variability such as hyperparameter details (Claesen and De Moor, 2015). We found that only 16 out of the 53 relevant papers had released their source code. 14 out of 53 papers were evaluated on multiple datasets, but the source code was publicly available for only five of those. Despite this, much of this code was lacking in modularity to be easily extendable to new datasets. In many cases, the process of reproducing the paper results was often unclear and lack of documentation made this more difficult. Even though most papers mentioned some hyperparameter details, important details were missing such as number of epochs, batch size, random initialization seed, if any, and details about early stop if that technique was applied.

Ablation Studies Lack of generalizability is caused by a dearth of appropriate empirical evaluation to identify the source of modeling gains. Ablation studies are important for identifying sources of improvements in results. Among the 53 papers that we looked at, 20 of the 24 papers in the general domain performed ablation studies. However, only 10 out of 29 papers in the medical domain performed one. Among these ablation studies,

---

[1] The 53 papers were filtered from a list of 728 papers skimmed for relevance. Appendix A contains paper details.

key details related to pre-processing were missing, which we found critical in our experiments.

In the absence of such information about causes of large variability of results, *fair comparison* of models becomes difficult. In this paper, we present an open-source unifying framework enabling the comparison of various training methodologies, pre-processing, modeling techniques, and evaluation metrics. The code is available at `https://github.com/geetickachauhan/relation-extraction`.

The experimental goals of this framework are identification of sources of variability in results for the three datasets and provide the field with a strong baseline model to compare against for future improvements. The design goals of this framework are identification of best practices for relation extraction and to be a guide for approaching new datasets.

By performing systematic comparison on three datasets, we find that 1) pre-processing choices can cause the largest variations in performance, 2) reporting scores on one test set split is problematic due to split bias. We perform other analyses in section 5 and also include recommendations for future research in this field in section 7.

Upon testing various combinations of our approaches, we achieve results near state of the art ranges for the three datasets: 85.89% macro F1 for Semeval 2010 task 8 dataset (Hendrickx et al., 2009) i.e. `semeval`, 71.97% macro F1 for DDI Extraction 2013 (Segura-Bedmar et al., 2013) i.e. `ddi` and 71.01% micro F1 for i2b2/VA 2010 relation classification dataset (Uzuner et al., 2011) i.e. `i2b2`. We refer to `ddi` and `i2b2` as medical datasets, as they belong to the biomedical and clinical domains, respectively.

Dataset	Rel	Eval	Agreement	Det
semeval	18	Macro	0.6-0.95	No
ddi	5	Macro	>0.8; 0.55-0.72	Yes
i2b2	8	Micro	-	Yes

Table 1: Dataset information, with columns Rel = number of relations, Eval = evaluation metric (all F1 scores), Agreement = Inter-annotator agreement, Det = whether detection task from section 3.4 was evaluated on. Rel column only includes relations used in official evaluation metric. `ddi` was built from two separately annotated sources and therefore contains two inter-annotator agreements.

2 Datasets

We summarize important information about these datasets in table 1. We introduce *detection* and *classification* tasks in section 3.4, but also indicate the tasks evaluated for each dataset in table 1.

Semeval 2010 `semeval` consists of 8000 training sentences and 2,717 test sentences for the multi-way classification of semantic relations between pairs of nominals. Not included in the official evaluation is an *Other* class which is considered noisy, with annotators choosing this class if no fit was found in the other classes. It is important to note that this is a synthetically generated dataset, and *detection* scores were not calculated due to the noisy nature of the *Other* class.

DDI Extraction `ddi` consists of 1,017 texts with 18,491 pharmacological substances and 5,021 drug-drug interactions from Pubmed articles in the pharmacological literature. *None* class indicating no interaction between the drug pairs is included in the evaluation metric calculation.

i2b2/VA 2010 relations `i2b2` consists of discharge summaries from Partners Healthcare and the MIMIC II Database (Saeed et al., 2011). They released 394 training reports, 477 test reports and 877 unannotated reports. After the challenge, only a part of the data was publicly released for research. *None* relation was present in the data and not considered in the official evaluation.

3 Methodology

Our framework breaks up processing into different stages, allowing for future modular addition of components. First, a `formatter` converts the raw dataset into a common comma separated value (CSV) input format accepted by the `pre-processor`, and this information is then fed to the `model`, which performs the training, after which `evaluation` is performed on the test set. With our framework, we test the following variations in the main components:

3.1 Pre-Processing

We test various pre-processing methods after performing simple tokenization and lower-casing of the words: entity blinding used by Liu et al. (2016b), stop-word and punctuation removal, and digit normalization commonly applied for `ddi` in (Zhao et al., 2016), and named entity recognition

related replacement (we call this NER blinding). We used the spaCy framework[2] for tokenization and to identify punctuation and digits.

Entity blinding and NER blinding are similar concept blinding techniques where the first is performed based on gold standard annotations, while the second is performed by running NER on the original sentence. We replace the words in the sentence matching the entity or named entity span with the target label and use those for training and testing.

Entity labels for `semeval` were not annotated with type information, whereas `ddi` identified drugs and `i2b2` identified medical problems, tests and treatments. Therefore, entity labels for `semeval` were *ENTITY*, for `ddi` were *DRUG* and for `i2b2` were *PROBLEM*, *TREATMENT* and *TEST*. In this paper, we use *fine-grained concept type* to refer to the presence of more than one concept type, as in the the case of `i2b2`.

NER labels for `semeval` consisted of those provided by the large english model by spaCy and provided standard types such as *PERSON* and *ORGANIZATION*, whereas those for the medical datasets was provided by the scispacy medium size model[3] and did not provide types. In this case, blinding consisted of replacing the words in the sentence by *Entity*.

We chose the spaCy model for NER to complement the extendable design goals of `REflex`. Other options such as cTAKES (Savova et al., 2010) for clinical data and MetaMAP[4] for biomedical data are highly specific to the dataset type and require running additional scripts outside of the `REflex` pipeline.

3.2 Modeling

We employ a baseline model based upon (Zeng et al., 2014), (Santos et al., 2015) and (Jin et al., 2018), which is a convolutional neural network (CNN) with position embeddings and a ranking loss (referred to as `CRCNN` in this paper). We initialize the model with pre-trained word embeddings: the senna embeddings by Collobert et al. (2011) for the general domain dataset and the `PubMed-PMC-wikipedia` embeddings released by Pyssalo et al. (2013) for the medical domain. We test several perturbations on top of `CRCNN` model, such as piecewise max-pooling, as

[2] https://github.com/explosion/spaCy
[3] https://allenai.github.io/scispacy/
[4] https://metamap.nlm.nih.gov

suggested by Zeng et al. (2015) and the more recent ELMo embeddings by Peters et al. (2018). To compare different featurizations of contextualized embeddings, we also employ the embeddings generated by the BERT model (rather than the standard fine-tuning approach). For ELMo, we use the Original (5.5B) model weights in `semeval` and PubMed contributed model weights in the medical datasets released by (Peters et al., 2018). For BERT, we use the BERT-large uncased model (without whole word masking) in `semeval` released by (Devlin et al., 2018), BioBERT by (Lee et al., 2019) in `ddi` and Clinical BERT by (Alsentzer et al., 2019) in `i2b2`.

The fine-tuning approach, which tends to be computationally expensive, has been thoroughly explored for multiple tasks, including medical relation extraction by Lee et al. (2019), but the approach of featurizing them with an existing model has not been explored in the literature as much. We tested different ways of featurizing the BERT contextualized embeddings for researchers who want to utilize a less computationally intensive technique, while still aiming for performance gains for their task.

Because ELMo provides token level embeddings, we chose to concatenate them with the word and position embeddings from `CRCNN` before the convolution phase. However, BERT provides word-piece level as well as sentence level embeddings. The first was concatenated similar to ELMo (which we call BERT-tokens), while the second was concatenated with the fixed size sentence representation outputted after convolution of word and position embeddings (BERT-CLS).

3.3 Training

We explore two ways of doing hyperparameter tuning: manual tuning and random search (Bergstra and Bengio, 2012).

Evaluating on three datasets meant that we needed to identify a default list of hyperparameters by tuning on one of the datasets before we could identify the hyperparameter list for the other two. We chose `semeval` for initial tuning due to its larger literature and because the `CRCNN` model was originally evaluated on this dataset. We started with reference hyperparameters listed in Zeng et al. (2014) and Santos et al. (2015) and identified default hyperparameters after tuning on a dev set randomly sampled from the training data

of the `semeval` dataset. These default hyperparameters[5] were used as starting points for manual tuning on the medical datasets as well as random search for all datasets.

We perform manual tuning on a subset of the hyperparameters, mentioned in table 2. In order to avoid overfitting in cross validation pointed out by Cawley and Talbot (2010), we perform a nested cross validation procedure, keeping a dev fold for hyperparameter tuning and a held out fold for score reporting.

On these dev folds, we perform paired t-tests for each of the perturbations to the parameters listed in table 2. Our first pass involves changing one hyperparameter per experiment and noting the ones that cause a statistically significant improvement, which helps us identify a narrower list of hyperparameters to tune on. We further refine the hyperparameter values in our second pass by testing on values similar to those that were leading to statistically significant improvements in the first pass. For example, if we noticed that lower epoch values were helpful in the first pass, we tested them in combination with the other optimal hyperparameter values (from first pass) in the second pass.

For each of the datasets, we tuned based on their official challenge evaluation metrics listed in section 2. `ddi` and `i2b2` had 5-fold nested cross validation performed on them, whereas `semeval` had 10-fold cross validation performed.

Random search was performed based on the official evaluation metrics for each dataset, on a fixed dev set randomly sampled from the training data. Final distributions are listed in table 3.

3.4 Evaluation

The official challenge problems for all datasets compared models based on multi-class classification, but for the medical datasets, we were also interested in looking at the changes in model performance if we treated the task as a binary classification problem. This was based on the rationale that in the drug literature, for example, pharmacologists would not want to sacrifice the ability to identify a potentially life threatening drug interaction pair, even if the type of the drug pair is not known. Therefore, we report results for both multi-class and binary classification scenarios. For clarity, we refer to them in the rest of the paper as *classification* and *detection* respectively.

Detection results were obtained using our evaluation scripts by treating existing relations as one class, ignoring the types outputted by the model. The other class in this task was the *None* or *Other* class, representing non-existing relations. Note that we did not re-train our model for this.

In addition to evaluating on two tasks for the medical and one task for the general dataset, we comment on the implications of different evaluation metrics in section 5.5.

4 Results

For experiments on the medical datasets i.e. `i2b2` and `ddi`, we used hyperparameters found from manual search individually performed on them. `semeval` had the default hyperparameters used for its experiments. These sets of hyperparameters were used in all experiments other than those reported in table 6, where we compare hyperparameter tuning methodologies.

Once we had a fixed set of hyperparameters for each dataset, we tested the perturbations for preprocessing as well as modeling in tables 4 and 5. Perturbations on the hyperparameter search are listed in table 6 and compare performance with different hyperparameter values found using different tuning strategies.

We generate the standard *classification* and the additional *detection* scores by the procedure described in section 3.4, and report these results under the *Class* and *Detect* columns.

We also report additional experiments in tables 7 and 8 based on the improvements found in tables 4 and 5. For all results tables, we report official test set results at the top, with accompanying cross validated results (averaged over all folds with their standard deviation) in smaller font below them.[6]

5 Discussion

Recently, CNNs have achieved strong performance for text classification and are typically more efficient than recurrent architectures (Bai et al., 2018; Kalchbrenner et al., 2014; Wang et al., 2015; Zhang et al., 2015b). The speed of our baseline `CRCNN` model allows us to explore multiple alternatives for every stage of our pipeline. We discuss these results pertaining to the *classification* task for all datasets and the *detection* task for

---

[5]listed in source code

[6]Results tables for metrics other than the official ones were omitted in the interest of space, but their analysis exists in section 5.5.

Hyperparameter	Values
epoch	{50,100,150,200}
lr decay	[1e-3, 1e-4, 1e-5]
sgd momentum	{T, F}
early stop	{T, F}
pos embed	{10, 50, 80, 100}
filter dimension	{50, 150}
filter size	2-3-4, 3-4-5
batch size	{70, 30}

Table 2: Hyperparameters explored for the first pass of manual search. lr decay means learning rate decay at [60, 120] epochs, pos embed refers to the position embedding size.

Hyperparameter	Distributions
epoch	uniform(70, 300)
lr	{constant, decay}
lr init	uniform(1e-5, 0.001)
filter size	2-3, 2-3-4, 2-3-4-5 3-4-5, 3-4-5-6
early stop	{T, F}
batch size	uniform(30, 70)

Table 3: Hyperparameter distributions for random search. Those written in {} are picked with equal probabilities. The learning rate (lr) was uniformly initialized, and decayed from 0.001 to the intialized value at half of the number of epochs. If early stop was true, patience was set to a fifth of the number of epochs. We ran 100-120 experiments for each dataset to search for optimal hyperparameters.

the medical datasets.

5.1 Pre-processing

Often, papers fail to mention the importance of pre-processing in performance improvements. Experiments in table 4 reveal that they can cause larger variations in performance than modeling.

We applied pre-processing changes with the `CRCNN` model with default hyperparameters for `semeval` and manual hyperparameters for the medical datasets. All comparisons are performed against the original pre-processing technique, which involved using the original dataset sentences in training and test.

Punctuation and digits hold more importance for the `ddi` dataset, which is a biomedical dataset, compared to the other two datasets. We looked at examples where this technique led to an incorrect prediction, but original pre-processing led to a correct one to investigate the source of performance further. The examples indicate that removal of punctuation is driving worse performance compared to the normalization of digits. A detailed analysis for these is present in (Chauhan, 2019).

Stop word removal is a common technique in Natural Language Processing (NLP) to simplify the sentence by cutting out commonly used words such as *the* and *is* in order to simplify the sentence. We found that stop words seem to be important for relation extraction for all three datasets that we looked at, to a smaller degree for `i2b2` compared to the other two datasets. Looking at examples misclassified by this technique revealed important stop words for different relations, which indicates that the removal of stop words is not beneficial in the relation extraction setting. Example types are shown in (Chauhan, 2019).

The availability of fine-grained concept types is likely to boost performance in relation extraction settings. The `i2b2` dataset provided fine-grained concept types in the form of medical problem, test and treatments. Entity blinding causes almost 9% improvement in *classification* performance and 1% improvement in *detection* performance. In contrast, `ddi` only provided gold standard annotations for drug types in the sentence, and while this does not cause statistically significant improvements for cross validation, it does improve test set classification performance by about 1.5% and detection performance by 1%. For these medical datasets, NER blinding consisted of replacing the detected named entities by *Entity* because named entity types were not available. Due to the coarse-grained nature of the entities, it hurts *classification* performance significantly, and *detection* performance a little.

While entity blinding hurts performance for `semeval`, possibly due to the coarse-grained nature of the replacement, NER blinding does not hurt performance. Looking at misclassified examples for entity blinding and NER blinding techniques supports this hypothesis (Chauhan, 2019).

To recall, entity blinding involved replacement of entity words by *Entity*, while NER blinding involved replacing named entities in the sentence with labels such as *ORGANIZATION* and *PERSON*. In settings where fine-grained entity blinding may not be helping, they may be helpful as added features into the model, as shown by (Socher et al., 2012).

For the medical datasets, while *classification* performance varies highly with different pre-

Dataset	semeval	ddi		i2b2	
Preprocess		Class	Detect	Class	Detect
Original	**81.55**	65.53	81.74	59.75	83.17
	80.85 (1.31)	82.23 (0.32)	88.40 (0.48)	70.10 (0.85)	86.45 (0.58)
Entity Blinding	72.73	**67.02**	**82.37**	**68.76**	**84.37**
	71.31 (1.14)	83.56 (2.05)•	89.45 (1.05)•	76.59 (1.07)	88.41 (0.37)
Punct and Digit	81.23	63.41	80.49	58.85	81.96
	80.95 (1.21)•	80.44 (1.77)	87.52 (0.98)	69.37 (1.43)•	85.82 (0.43)
Punct, Digit and Stop	72.92	55.87	76.57	56.19	80.47
	71.61 (1.25)	78.52 (1.99)	85.65 (1.21)	68.14 (2.05)•	84.84 (0.77)
NER Blinding	81.63	57.22	79.03	50.41	81.61
	80.85 (1.07)•	78.06 (1.45)	86.79 (0.65)	66.26 (2.44)	86.72 (0.57)•

Table 4: Pre-processing techniques with `CRCNN` model. Row labels Original = simple tokenization and lower casing of words, Punct = punctuation removal, Digit = digit removal and Stop = stop word removal. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to Original pre-processing ($p < 0.05$) using a paired t-test except those marked with a •

Dataset	semeval	ddi		i2b2	
Modeling		Class	Detect	Class	Detect
CRCNN	81.55	65.53	81.74	59.75	83.17
	80.85 (1.31)	82.23 (0.32)	88.40 (0.48)	70.10 (0.85)	86.45 (0.58)
Piecewise pool	81.59	63.01	80.62	60.85	83.69
	80.55 (0.99)•	81.99 (0.38)•	88.47 (0.48)•	73.79 (0.97)	89.29 (0.61)
BERT-tokens	85.67	**71.97**	**86.53**	63.11	**84.91**
	85.63 (0.83)	85.35 (0.53)	90.70 (0.46)	72.06 (1.36)	87.57 (0.75)
BERT-CLS	82.42	61.3	79.63	56.79	81.91
	80.83 (1.18)•	82.71 (0.68)•	88.35 (0.77)•	67.37 (1.08)	85.43 (0.36)
ELMo	**85.89**	66.63	83.05	**63.18**	84.54
	84.79 (1.08)	84.53 (0.96)	90.11 (0.56)	72.53 (0.80)	87.81 (0.34)

Table 5: Modeling techniques with original pre-processing. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to `CRCNN` model ($p < 0.05$) using a paired t-test except those marked with a •. In terms of statistical significance, comparing contextualized embeddings with each other reveals that BERT-tokens is equivalent to ELMo for `i2b2`, but for `semeval` BERT-tokens is better than ELMo and for `ddi` BERT-tokens is better than ELMo only for detection.

processing techniques, *detection* is relatively unaffected. In a setting where one cares more about detection of relationships rather than multi-class classification, one would be able to get away with using non-complicated pre-processing techniques to maintain reasonable performance.

5.2 Split Bias

All three datasets evaluate models based on one score on the test set, which is common practice for NLP challenges. Reporting one score as opposed to a distribution of scores has been shown to be problematic by Reimers and Gurevych (2017) for sequence tagging. Recently, Crane (2018) discuss similar problems for question-answering. We show that even if you keep the same random initialization seed (all our experiments have a fixed random initialization seed), split bias can be another source of variation in scores.

In our experiments, significance testing of some cross validated results reveals no significance even when the test set result improves in performance. This is particularly concerning for `ddi` where entity blinding (called drug blinding in the literature) is used as a standard pre-processing technique without ablation studies demonstrating its effectiveness. Our results suggest the contrary: entity blinding seems to help test set performance for `ddi` in table 4, but shows no statistical significance. Table 8 further demonstrates that using this in conjunction with other techniques results in test score variations despite being statistically insignif-

| Dataset | semeval | ddi | | i2b2 | |
Hyperparam Tuning		Class	Detect	Class	Detect
Default	81.55	62.55	80.29	55.15	81.98
	80.85 (1.31)	81.62 (1.35)	87.76 (1.03)	67.28 (1.83)	86.57 (0.58)
Manual Search	-	**65.53**	**81.74**	**59.75**	**83.17**
		82.23 (0.32)•	88.40 (0.48)•	70.10 (0.85)	86.45 (0.58)•
Random Search	**82.2**	62.29	79.04	55.0	80.77
	81.10 (1.26)•	75.43 (1.48)	83.54 (0.60)	60.66 (1.43)	82.73 (0.49)

Table 6: Hyperparameter tuning methods with original pre-processing and fixed `CRCNN` model. Test set results at the top with cross validated results (average with standard deviation) below. All cross validated results are statistically significant compared to Default with $p < 0.05$ except those marked with a •. Note that hyperparameter tuning can involve much higher performance variation depending on the distribution of the data. Therefore, even though there is no statistical significance in the manual search case for the held out fold in the ddi dataset, there was statistical significance for the dev fold which drove those set of hyperparameters. For both ddi and i2b2 datasets, manual search is better than random search with $p < 0.05$.

| Task | Classification | Detection |
Technique		
E + ent	70.46	86.17
	77.70(1.26)	89.36 (0.50)
B + ent	70.56	85.66
	76.72 (1.04)	88.63 (0.33)
E + piece + ent	70.62	86.14
	79.41 (0.53)	90.37 (0.44)
B + piece + ent	**71.01**	**86.26**
	79.51 (1.09)	90.34 (0.53)
piece + ent	69.73	85.44
	78.12 (1.10)	89.74 (0.44)
E + piece	63.19	84.92
	74.76 (0.68)	89.90 (0.37)
B + piece	63.23	85.45
	74.67 (0.89)	89.61 (0.68)

Table 7: Additional experiments for `i2b2`. E = ELMo, B = BERT-tokens, ent = entity blinding, piece = piecewise pooling. All results are statistically significant compared to BERT-tokens and ELMo models respectively from table 5 and piece + ent row is statistically significant compared to piecewise pool model as well as entity blinding model. These are all statistically significantly better than the `CRCNN` model from table 5.

| Task | Classification | Detection |
Technique		
E + ent	68.69	83.72
	86.25 (1.54)	91.35 (0.90)
B + ent	**70.66**	**85.35**
	85.79 (1.54)	91.26 (0.63)

Table 8: Additional experiments for `ddi`. E = ELMo, B = BERT-tokens, ent = entity blinding. Results are not statistically significant compared to BERT-tokens and ELMo models respectively from table 5 and not from each other either.

icant.

No statistical significance is seen even when the test set result worsens in performance for BERT-CLS in table 5 where it hurts test set performance on `ddi` but is not statistically significant when cross validation is performed.

5.3 Modeling

In table 5, we tested the generalizability of the commonly used piecewise pooling technique proposed in (Zeng et al., 2015), a variant of which was applied in the model by Luo et al. for `i2b2`. We also tested the improvements offered by different featurizations of contextualized embeddings, which has not been explored much for relation extraction.

Modeling changes were applied with the original pre-processing technique for the `CRCNN` model with default hyperparameters for `semeval` and manual hyperparameters for the medical datasets. All comparisons are performed with the baseline performance of the `CRCNN` model.

While piecewise pooling helps `i2b2` by 1%, it hurts test set performance on `ddi` and doesn't affect performance on `semeval`. While it may be intuitive to split pooling by entity location, this technique is not generalizable to other datasets.

We also found that while contextualized embeddings generally boost performance, they should be concatenated with the word embeddings before the convolution stage to cause a significant boost in performance. We found ELMo and BERT-tokens to boost performance significantly for all

datasets, but that BERT-CLS hurt performance for the medical datasets. While BERT-CLS boosted test set performance for `semeval`, this was not found to be a statistically significant difference for cross validation. Note that we featurized ELMo similarly to BERT-tokens and the details are present in section 3.2.

This indicates that the technique of featurizing the contextualized embeddings is important for a CNN architecture. Concatenating the contextualized embeddings with the word embeddings keeps a tighter coupling, which is helpful for relation extraction where the word level associations are essential in predicting the relation type.

5.4 Hyperparameter Tuning

Bergstra and Bengio (2012) show the superiority of random search over grid search in terms of faster convergence, but leave to future work automating the procedure of manual tuning, i.e. sequential optimization. Bayesian optimization strategies could help with this (Snoek et al., 2012) but often require expert knowledge for correct application. We tested how manual tuning, requiring less expert knowledge than Bayesian optimization, would compare to the random search strategy in table 6. For both `i2b2` and `ddi` corpora, manual search outperformed random search.

5.5 Evaluation Metrics

Picking the right evaluation metric for a dataset is critical, and it is important to choose a metric that has the biggest delta between different model performances for example types we care about. Tables for different metric results for all datasets are provided in Appendix B.

When using micro and macro statistics (precision, recall and F1), class imbalance dictates the one to pick. Macro statistics are highly affected by imbalance, whereas micro statistics are able to recover well. Despite suffering due to class imbalance, though, macro statistics may be more appropriate than micro as they provide stronger discriminative capabilities by providing equal importance to classes of smaller sizes. However, micro statistics are as discriminative as macro statistics in settings when the classes are relatively balanced. We are going to talk about the *classification* tasks in the next two paragraphs.

Compared to `semeval`, `ddi` and `i2b2` suffer from stark class imbalances. `semeval` has a

number of examples in non-*Other* classes ranging from 200 or 300 to 1000. *Other* class has about 3000 examples which are not included in the official metric calculations. `ddi` has one class with 228 examples, while the others have about 1000 examples. The *None* class has 21,948 examples which is included for the official score calculations. `i2b2` has five classes in the 100-500 range, while the others contain about 2000 examples. *None* is the largest class with 19,934 examples.

Using micro statistics is reasonable for `i2b2` because the highly imbalanced class is not included in the calculations. Therefore, this metric is able to be as discriminative as macro statistics. For example, test set micro F1 between baseline and entity blinding techniques is 59.75 and 68.76, while that for macro F1 is 36.44 and 43.76. In contrast, using micro statistics is a bad idea for `ddi` because the performance on the *None* class would drive most of the predictive results of the model. For example, micro-F1 between baseline and NER blinding is 88.69 and 86.18, whereas macro-F1 is 65.53 and 57.22. `semeval` does not have a stark contrast between micro and macro scores due to *Other* class not being included in the calculation. Using either metric to evaluate models is reasonable for this dataset.

The detection task does not suffer from such variations due to the lower class imbalance. For example, `ddi` dataset micro-F1 between baseline and NER blinding model is 90.01 and 88.74, while macro-F1 is 81.74 and 79.03. This further suggests that modeling differences and pre-processing differences cause more variation in performance in settings when the class imbalance is higher.

6 Comparison with SOTA

The best *classification* test set results found are listed in table 9. Note that we do not compare the *extraction* task for datasets other than `ddi` because the official challenges only compared *classification* results. Even though the official challenge did not rank models based on the *detection* task, recent papers in the `ddi` literature mention these results.

Wang et al. (2016) report a result of 88% on `semeval` and do not provide any public source code for replication purposes. Despite being below the state of the art range, `REflex` provides the best performing publicly available model for

Dataset	Result	Technique
semeval	85.89	E
ddi	71.97, 86.53	B
i2b2	71.01	B + piece + ent

Table 9: Best test set *classification* results for all datasets, except `ddi` where *detection* results are mentioned after the classification results. piece = Piecewise pooling, ent = entity blinding, E = ELMo, B = BERT-tokens. Result corresponds to F1 scores, macro for `semeval` and `ddi`, but micro for `i2b2`.

this dataset. Zheng et al. (2017) report the best result on `ddi` (77.3%) but perform negative instance filtering, which is a highly specific pre-processing technique that does not fit with the flexible nature of `REflex`. This technique cuts specific examples from the dataset, but the paper is unclear about whether train as well as test data are shortened. If the test data is being shortened, the performance comparison becomes unfair due to evaluation on different test samples. Unfortunately, source code was not publicly available to answer these questions.

Note that Zhao et al. (2016) show that negative instance filtering causes a 4.1% improvement in test set performance. If `REflex` were to use this pre-processing technique, it would reach close to the state-of-the-art (SOTA) number on the *classification* task. On the other hand, results from the *detection* results *outperform* this model by 2.53%.

Sahu et al. (2016) (code unavailable) report a state of the art result of 71.16% on `i2b2`, which the results in table 9 are able to match. Note that (Rink et al., 2011) report a result of 73.7% with a support vector machine, but they used a larger version of the dataset. Comparison against different subsets of the dataset would not be fair.

Comparison against these numbers demonstrates that `REflex` is the only open-source framework, providing performance near SOTA ranges for the three datasets. Therefore, `REflex` can be used as a strong baseline model in future relation extraction studies.

7 Conclusion

Our findings reveal variations offered by pre-processing and training methodologies, which often go unreported. They indicate that comparing models without having these techniques standardized can make it difficult to assess the true source of performance gains. Our key findings are:

1. Pre-processing can have a strong effect on performance, sometimes more than modeling techniques, as is the case of `i2b2`. Concept types seem to offer useful information, perhaps revealing more general semantic information in the sentence that can help with predictions. Fine-grained Gold standard annotated concept types are most beneficial, but those from automatically extracted packages may also be useful as long as they consist of multiple types. Punctuation and digits may hold more importance in biomedical settings, but stop words hold significance in all settings.

2. Reporting on one test set score can be problematic due to split bias, and a cross validation approach with significance tests may help ease some of this bias. Drug blinding for `ddi` is commonly used in the literature but does not seem to offer any statistically significant improvements. Therefore, it is unnecessary to use in this domain.

3. Contextualized embeddings are generally helpful but the featurizing technique is important: for CNN models, concatenating them with the word embeddings before convolution is most beneficial.

4. Picking the right hyperparameters for a dataset is important to performance. We suggest an initial manual hyperparameter search based on cross validation significance tests because that may be sufficient in most cases. If one is not pressed for time, random search is a reasonable automated option for hyperparameter tuning, but requires more experience for picking the right search space and the right distributions for the hyperparameters.

5. Picking the right evaluation metrics for a new dataset should be driven by class imbalance issues for the classes chosen to be evaluated on.

Acknowledgments

References

Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 828–838. Association for Computational Linguistics.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 756–765.

Gavin C Cawley and Nicola LC Talbot. 2010. On overfitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107.

Geeticka Chauhan. 2019. *REflex: Flexible Framework for Relation Extraction in Multiple Domains*. Master's thesis, Massachusetts Institute of Technology.

Veera Raghavendra Chikka and Kamalakar Karlapalem. 2018. A hybrid deep learning approach for medical relation extraction. *CoRR*.

Marc Claesen and Bart De Moor. 2015. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Matt Crane. 2018. Questionable answers in question answering research: Reproducibility and variability of published results. *Transactions of the Association of Computational Linguistics*, 6:241–252.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Javid Ebrahimi and Dejing Dou. 2015. Chain based rnn for relation classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1244–1249.

Antske Fokkens, Marieke Van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1691–1701.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18. Association for Computational Linguistics.

Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1372–1376.

Bin He, Yi Guan, and Rui Dai. 2018a. Classifying medical relations in clinical text via convolutional neural networks. *Artificial intelligence in medicine*.

Bin He, Yi Guan, and Rui Dai. 2018b. Convolutional gated recurrent units for medical relation classification. *CoRR*, abs/1807.11082.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Degen Huang, Zhenchao Jiang, Li Zou, and Lishuang Li. 2017. Drug-drug interaction extraction from biomedical literature using support vector machine and long short term memory networks. *Information Sciences*, 415.

Darrel C Ince, Leslie Hatton, and John Graham-Cumming. 2012. The case for open computer programs. *Nature*, 482(7386):485.

Di Jin, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. 2018. MIT-MEDG at SemEval-2018 Task 7: Semantic Relation Classification via Convolution Neural Network. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 798–804. Association for Computational Linguistics.

Wen Juan Hou and Bamfa Ceesay. 2018. Extraction of drug-drug interaction using neural embedding. *Journal of Bioinformatics and Computational Biology*, 16.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics.

Ramakanth Kavuluru, Anthony Rios, and Tung Tran. 2017. Extracting drug-drug interactions with word and character-level recurrent neural networks. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 5–12.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342. Association for Computational Linguistics.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*.

Q. Li, Z. Yang, L. Luo, L. Wang, Y. Zhang, H. Lin, J. Wang, L. Yang, K. Xu, and Y. Zhang. 2018a. A multi-task learning based approach to biomedical entity relation extraction. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 680–682.

Yifu Li, Ran Jin, and Yuan Luo. 2018b. Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns). *Journal of the American Medical Informatics Association*, 26(3):262–268.

Sangrak Lim and Jaewoo Kang. 2018a. Chemicalgene relation extraction using recursive neural network. In *Database*.

Sangrak Lim and Jaewoo Kang. 2018b. Drug drug interaction extraction from the literature using a recursive neural network. In *PloS one*.

Sangrak Lim, Kyubum Lee, and Jaewoo Kang. 2018. Drug drug interaction extraction from the literature using a recursive neural network. *Plos one*, 13:1–17.

Shengyu Liu, Kai Chen, Qingcai Chen, and Buzhou Tang. 2016a. Dependency-based convolutional neural network for drug-drug interaction extraction. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1074–1080.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2016b. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016.

Yuan Luo. 2017. Recurrent neural networks for classifying relations in clinical notes. *Journal of Biomedical Informatics*, 72.

Yuan Luo, Yu Cheng, Özlem Uzuner, Peter Szolovits, and Justin Starren. 2017. Segment convolutional neural networks (seg-cnns) for classifying relations in clinical notes. *Journal of the American Medical Informatics Association*, 25(1):93–98.

Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. 2016. Clinical relation extraction with deep learning. In *International Journal of Hybrid Information Technology*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2018. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. *arXiv preprint arXiv:1805.10586*.

Thien Huu Nguyen and Ralph Grishman. 2015. Combining neural networks and log-linear models to improve relation extraction. *CoRR*, abs/1511.05926.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Sampo Pyssalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43.

Pengda Qin, Weiran Xu, and Jun Guo. 2016. An empirical convolutional neural network approach for semantic relation classification. *Neurocomput.*, 190:1–9.

Changqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. 2016. Multichannel convolutional neural network for biological relation extraction. In *BioMed research international*.

40

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Bryan Rink, Sanda Harabagiu, and Kirk Roberts. 2011. Automatic extraction of relations between medical concepts in clinical texts. *Journal of the American Medical Informatics Association*, 18(5):594–600.

Jonathan Rotsztejn, Nora Hollenstein, and Ce Zhang. 2018. Eth-ds3lab at semeval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 689–696. Association for Computational Linguistics.

Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. 2011. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952.

Sunil Sahu, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu. 2016. Relation extraction from clinical texts using domain invariant convolutional neural network. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 206–215. Association for Computational Linguistics.

Sunil Kumar Sahu and Ashish Anand. 2018. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15 – 24.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Isabel Segura-Bedmar, Paloma Martínez, and María 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 (SemEval 2013)*, volume 2, pages 341–350.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics.

Simon Suster, Madhumita Sushil, and Walter Daelemans. 2018. Revisiting neural relation classification in clinical notes with external information. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 22–28. Association for Computational Linguistics.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539. Association for Computational Linguistics.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, volume 1, pages 1298–1307.

Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 352–357.

Wei Wang, Xi Yang, Canqun Yang, Xiao-Wei Guo, Xiang Zhang, and Chengkun Wu. 2017. Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinformatics*, 18.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks

with data augmentation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470. The COLING 2016 Organizing Committee.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *CoRR*, abs/1702.01923.

Mo Yu, Matthew Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *NIPS Workshop on Learning Semantics*, pages 95–101.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015a. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics.

Zhehuan Zhao, Zhihao Yang, Ling Luo, Hongfei Lin, and Jian Wang. 2016. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, 32(22):3444–3453.

Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Yijia Zhang, Zhihao Yang, and Jian Wang. 2017. An attention-based effective neural model for drug-drug interactions extraction. In *BMC Bioinformatics*.

A   Quantitative Literature Review

paper	cite	code	ablation	hyperparam	cross val	word-embed	datasets
Socher et al. (2012)	890	y	•	y	•	y	2
Zeng et al. (2014)	477	•	y	y	y	y	1
Santos et al. (2015)	220	•	y	y	y	y	1
Nguyen and Verspoor (2018)	146	•	y	y	y	•	2
Miwa and Bansal (2016)	175	•	y	y	y	•	3
Li and Jurafsky (2015)	107	y	y	y	•	y	6
Xu et al. (2015a)	108	•	y	y	•	y	1
Wang et al. (2016)	102	•	y	•	•	y	1
Hashimoto et al. (2013)	64	•	y	y	•	y	1
Zhang and Wang (2015)	68	•	y	•	y	y	2
Vu et al. (2016)	57	•	y	y	•	y	1
Yin et al. (2017)	116	•	n	y	•	•	7
Yu et al. (2014)	45	y	y	y	y	y	1
Xu et al. (2016)	54	y	y	y	•	•	1
Zhang et al. (2015a)	51	•	•	•	•	y	1
Nguyen and Grishman (2015)	42	•	y	y	•	y	2
Qin et al. (2016)	39	•	•	y	y	y	1
Cai et al. (2016)	44	•	y	y	•	y	1
Sahu et al. (2016)	32	•	y	y	y	y	1
Adel et al. (2016)	29	y	y	•	•	y	1
Zeng et al. (2015)	190	•	y	y	•	y	1
Xu et al. (2015b)	171	•	y	y	•	y	1
Zhang et al. (2018)	3	•	y	y	•	y	2
Levy et al. (2017)	20	y	y	y	•	y	1
Liu et al. (2016b)	48	•	•	y	•	y	1
Zhao et al. (2016)	41	y	y	y	•	y	1
Ebrahimi and Dou (2015)	30	•	•	•	•	•	2
Li et al. (2017)	27	y	y	y	y	y	2
Quan et al. (2016)	23	y	•	y	y	y	2

43

Paper	cite	code	ablation	hyperparam	cross val	word-embed	datasets
Sahu and Anand (2018)	13	y	y	y	•	y	1
Liu et al. (2016a)	9	•	•	y	•	y	1
Lim and Kang (2018b)	4	•	•	•	•	•	1
Zheng et al. (2017)	12	•	y	y	y	y	1
Wang et al. (2017)	5	n	y	y	•	y	1
Lim et al. (2018)	1	y	y	y	y	y	2
Kavuluru et al. (2017)	8	•	•	y	•	•	1
Huang et al. (2017)	4	•	•	y	•	y	1
Juan Hou and Ceesay (2018)	1	•	•	•	•	y	1
Lim and Kang (2018a)	4	y	•	y	•	y	1
Rotsztejn et al. (2018)	2	•	•	y	y	y	1
Jin et al. (2018)	0	•	y	y	y	y	1
Sahu et al. (2016)	31	•	y	y	y	y	1
Luo (2017)	21	•	•	y	•	y	1
Lv et al. (2016)	15	•	•	•	•	•	1
Jin et al. (2018)	14	•	y	y	•	y	1
Chikka and Karlapalem (2018)	1	y	•	y	•	•	1
Li et al. (2018b)	0	y	•	y	y	y	1
Li et al. (2018a)	0	•	•	•	•	•	5
Suster et al. (2018)	0	y	•	y	•	y	1
Luo et al. (2017)	16	y	•	y	•	y	1
He et al. (2018a)	2	•	•	y	•	y	1
He et al. (2018b)	0	•	•	y	y	y	2
Nguyen and Verspoor (2018)	1	•	y	y	•	y	1

Table 10: Following are the columns in this table: **cite** = number of papers that cited the paper; **code** = whether code was publicly available (y for yes and • for no); **ablation** = whether an ablation study was performed; **hyperparam** = whether hyperparameter details were mentioned; **cross val** = whether cross validation details were mentioned; **word-embed** = whether information about word embeddings used was mentioned; **datasets** = number of datasets evaluated on

B   Evaluation Metric Results on Test Data

Each row represents a pre-processing, modeling technique or combination based on the additional experiments run on each dataset. Only test set results (as opposed to cross validation) are reported for ease of analysis. In all the tables, Baseline refers to the CRCNN model with original pre-processing and default hyperparameters for semeval and manual hyperparameters for the medical datasets (ddi and i2b2). The following short forms are used as row labels:

B = BERT-tokens

E = ELMo

Ent Blind = Entity Blinding

Piece Pool = Piecewise Pooling

Technique / Metric	acc	micro-P	micro-R	micro-F1	macro-P	macro-R	macro-F1
Baseline	77.11	79.95	85.11	82.45	79.25	84.06	81.55
Entity Blinding	67.94	70.72	77.15	73.8	69.77	76.31	72.73
Punct and Digit	76.48	79.19	85.42	82.19	78.33	84.51	81.23
Punct, Digit and Stop	68.28	73.0	74.78	73.88	72.84	73.48	72.92
NER Blinding	77.25	79.3	86.03	82.53	78.49	85.13	81.63
Piecewise pool	77.0	79.54	85.55	82.44	78.86	84.71	81.59
ELMo	77.77	81.87	84.62	83.22	81.24	83.71	82.42
BERT-CLS	77.77	81.87	84.62	83.22	81.24	83.71	82.42
BERT-tokens	81.3	86.63	86.74	86.69	86.08	85.61	85.67

Table 11: Different Evaluation Metric results on test set of semeval dataset. Only test set results are reported for ease of analysis. Metric short forms used are **acc** = accuracy; **P** = precision; **R** = recall.

Metric	acc		micro-P		micro-R		micro-F1		macro-P		macro-R		macro-F1	
Technique	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect
Baseline	88.69	90.01	88.69	90.01	88.69	90.01	88.69	90.01	72.32	82.06	63.48	81.43	65.53	81.74
Entity Blinding	89.22	90.44	89.22	90.44	89.22	90.44	89.22	90.44	71.26	82.99	64.63	81.79	67.02	82.37
Punct and Digit	88.31	89.61	88.31	89.61	88.31	89.61	88.31	89.61	69.49	81.7	60.81	79.43	63.41	80.49
Punct, Digit and Stop	86.58	87.86	86.58	87.86	86.58	87.86	86.58	87.86	67.4	78.59	52.72	74.98	55.87	76.57
NER Blinding	86.18	88.74	86.18	88.74	86.18	88.74	86.18	88.74	59.13	79.9	55.93	78.24	57.22	79.03
Piecewise pool	88.14	89.54	88.14	89.54	88.14	89.54	88.14	89.54	70.49	81.39	60.38	79.91	63.01	80.62
E	89.76	90.97	89.76	90.97	89.76	90.97	89.76	90.97	73.41	84.36	63.65	81.9	66.63	83.05
BERT-CLS	87.84	89.05	87.84	89.05	87.84	89.05	87.84	89.05	68.2	80.51	59.31	78.84	61.3	79.63
B	91.31	92.72	91.31	92.72	91.31	92.72	91.31	92.72	77.66	87.34	69.27	85.78	71.97	86.53
E + Entity Blinding	89.97	91.18	89.97	91.18	89.97	91.18	89.97	91.18	72.44	84.42	66.41	83.06	68.69	83.72
B + Entity Blinding	90.93	92.15	90.93	92.15	90.93	92.15	90.93	92.15	76.79	86.57	63.39	84.26	70.66	85.35

Table 12: Different Evaluation Metric results on test set of ddi dataset. Only test set results are reported for ease of analysis. Metric short forms used are **acc** = accuracy; **P** = precision; **R** = recall.

Metric / Technique	acc		micro-P		micro-R		micro-F1		macro-P		macro-R		macro-F1	
	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect	Class	Detect
Baseline	78.68	83.17	61.39	83.17	58.19	83.17	59.75	83.17	49.24	81.16	34.2	80.29	36.44	80.69
Entity Blinding	81.92	84.37	68.88	84.37	68.65	84.37	68.76	84.37	53.33	82.32	40.72	82.27	43.76	82.29
Punct and Digit	77.25	81.96	58.09	81.96	59.64	81.96	58.85	81.96	49.28	79.53	33.56	79.92	34.93	79.71
Punct, Digit and Stop	76.05	80.47	57.15	80.47	55.27	80.47	56.19	80.47	43.26	77.96	31.16	77.47	32.99	77.7
NER Blinding	75.12	81.61	52.58	81.61	48.42	81.61	50.41	81.61	39.44	79.45	26.3	78.17	29.15	78.73
Piecewise pool	78.63	83.69	59.41	83.69	62.37	83.69	60.85	83.69	46.16	81.41	35.77	82.17	36.44	81.76
E	80.4	84.54	64.56	84.54	61.86	84.54	63.18	84.54	59.28	82.69	36.17	81.97	38.1	82.31
BERT-CLS	76.94	81.91	57.66	81.91	55.95	81.91	56.79	81.91	49.88	76.61	32.4	79.15	34.05	79.37
B	80.79	84.91	64.92	84.91	61.4	84.91	63.11	84.91	58.05	83.08	36.8	82.1	39.31	82.55
E + Entity Blinding	83.62	86.17	72.43	86.17	68.6	86.17	70.46	86.17	60.79	84.65	40.11	83.67	42.99	84.13
E + Piece Pool + Ent Blind	83.46	86.14	71.11	86.14	70.14	86.14	70.62	86.14	54.87	84.37	42.41	84.13	44.43	84.25
Ent Blind + Piece Pool	82.72	85.44	69.49	85.44	69.98	85.44	69.73	85.44	48.82	83.49	41.97	83.61	42.89	83.55
E + Piece Pool	80.1	84.92	61.98	84.92	64.45	84.92	63.19	84.92	49.68	82.79	36.91	83.43	37.52	83.09
B + Ent Blind	83.27	85.66	71.52	85.66	69.63	85.66	70.56	85.66	55.62	83.9	38.82	83.44	41.83	83.66
B + Ent Blind + Piece pool	83.57	86.26	70.9	86.26	71.13	86.26	71.01	86.26	55.6	84.43	42.58	84.49	44.4	84.46
B + Piece pool	80.59	85.45	63.08	85.45	63.39	85.45	63.23	85.45	56.01	83.51	36.84	83.59	38.84	83.55

Table 13: Different Evaluation Metric results on test set of i2b2 dataset. Only test set results are reported for ease of analysis. Metric short forms used are **acc** = accuracy; **P** = precision; **R** = recall.