Cross-Domain Evaluation of Edge Detection for Biomedical Event Extraction

Alan Ramponi,[1,2,3] **Barbara Plank,**[2] **Rosario Lombardo**[3]
[1] Department of Information Engineering and Computer Science, University of Trento, Italy
[2] Department of Computer Science, IT University of Copenhagen, Denmark
[3] Fondazione the Microsoft Research – University of Trento Centre for Computational and Systems Biology, Italy
alan.ramponi@unitn.it, bplank@itu.dk, lombardo@cosbi.eu

Abstract

Biomedical event extraction is a crucial task in order to automatically extract information from the increasingly growing body of biomedical literature. Despite advances in the methods in recent years, most event extraction systems are still evaluated in-domain and on complete event structures only. This makes it hard to determine the performance of intermediate stages of the task, such as edge detection, across different corpora. Motivated by these limitations, we present the first cross-domain study of edge detection for biomedical event extraction. We analyze differences between five existing gold standard corpora, create a standardized benchmark corpus, and provide a strong baseline model for edge detection. Experiments show a large drop in performance when the baseline is applied on out-of-domain data, confirming the need for domain adaptation methods for the task. To encourage research efforts in this direction, we make both the data and the baseline available to the research community: `https://www.cosbi.eu/cfx/9985`.

Keywords: Corpus (Creation, Annotation, etc.), Information Extraction, Information Retrieval, Statistical and Machine Learning Methods.

1. Introduction

Information extraction systems in the biomedical field are valuable for a wide range of purposes, from the population of knowledge bases to the construction of biochemical pathways (Ananiadou et al., 2010). Among the natural language processing tasks for information extraction is **Event Extraction** (EE). Its goal is to extract semantically rich, structured information from unstructured texts. These representations, called *events*, are suitable to capture the elaborate biomedical statements in the scientific literature.[1] The expressivity of EE commonly comes at the cost of multiple classification stages (Figure 1). Given the input text and entity annotations, these stages are: (1) *trigger detection*: the identification of words – usually verbs or nominalized verbs – that may trigger events, and the assignment of the event type they express; (2) *edge detection*: the identification and classification of the semantic relations which hold between trigger words and named entities (or other triggers); and (3) *event construction*: the building of complete, multi-argument event structures from the edges. For instance, consider the example in Figure 1. Firstly, "phosphorylation" and "augments" are identified as `Phosphorylation` and `+Regulation` triggers, respectively. Then, arguments for those event triggers are determined (e.g., `Phosphorylation` is the `Cause` of `+Regulation`). Lastly, arguments are composed into self-contained event structures: two `Phosphorylation` events and two `+Regulation` events.

Recent studies on EE have shown that supervised machine learning approaches and in particular neural methods provide state-of-the-art performance on the task (Björne and Salakoski, 2018; Li et al., 2019). These methods require labeled data, therefore are trained and evaluated using corpora that have been manually annotated by field experts mainly in the context of community challenges (i.e., Shared Tasks) (Huang and Lu, 2016).

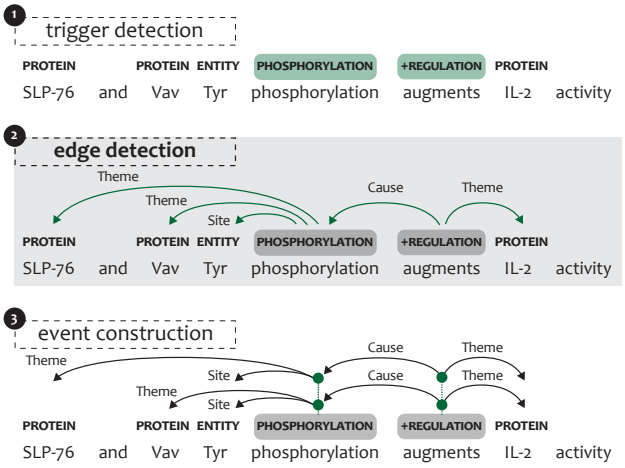

Figure 1: The classification stages in EE for the example sentence "*SLP-76 and Vav Tyr phosphorylation augments IL-2 activity*". The responsibilities of each stage are marked in *green*. In this work, we focus on the edge detection stage (illustrated within a *grey* box).

Despite the progress in the techniques, most EE systems are developed and evaluated under a strong, closed-world assumption which hinders their application in real-world scenarios. In fact, current models are typically trained under the implicit hypothesis that the test data (i.e., the *target*) follows the same underlying distribution of the training data (i.e., the *source*), an assumption that is clearly violated in the real world. In practice, this translates to a dramatic drop in performance when the model is applied – or evaluated – *into the wild* (i.e., *out-of-domain*), due to the differences of

[1] Unlike traditional relation extraction, event representations can capture the association of one or more participants in different semantic roles, where each association in turn can be argument of higher-level associations.

source and target corpora. This is the case of biomedicine, a field that is often seen as a domain per se, but instead comprises a lot of sub-domains, from molecular biology to genetics and physiology. In order to adequately assess the performance of biomedical EE systems, a thorough cross-domain evaluation is needed.

Since biomedical EE is typically framed as a multi-stage task, this evaluation could be carried out at each stage, similarly to what happens in the non-biomedical ACE event extraction challenge (Walker et al., 2006). However, biomedical EE has traditionally given emphasis on the equality of complete event structures only, for which predicted results are submitted to Shared Task online evaluation services. This has three main consequences: (i) test data is blind and is meant for the evaluation of entire events, thus it cannot be used for intermediate stages; (ii) most work report the final results only, making it difficult to evaluate and interpret how well the stages perform in isolation; and (iii) even if those performances are reported, results are incomparable due the different preprocessing conditions, such as the generation of negative examples, and experimental setups.

In this work, we focus on the edge detection stage since we believe it represents the most important module of the EE pipeline. In fact, in addition to being the middle step where both input and output data are not explicitly available, we argue the task shares most of the incomparability issues with relation extraction, including the number of negative examples one could generate, and the independence of training and test data with regard to the examples within the same sentences (Pyysalo et al., 2008).

Contributions. In the absence of explicit data and common means to evaluate edge detection in biomedical EE, we contribute to the field and provide:

- Standardized training and test data for edge detection for five different gold-standard corpora enabling cross-domain experimentation, together with a characterization of the differences between the corpora;
- A model for edge detection based on recent advances in neural methods, setting it as a strong baseline for future research;
- A thorough experimentation of edge detection in a cross-domain setting, quantifying the drop in performance of baseline models.

To the best of our knowledge, we are the first to provide such insights. We thus believe our work could encourage research efforts in domain adaptation in the near future, as well as in-depth evaluation of other stages.

2. Related Work

In recent years, a number of Shared Tasks have been organized in order to promote the development of techniques for biomedical natural language processing, providing annotated corpora and evaluation means (Huang and Lu, 2016). Of particular interest for biomedical EE are the GE11 corpus (Kim et al., 2011), the ID11 corpus (Pyysalo et al., 2011), the EPI11 corpus (Ohta et al., 2011), the PC13 corpus (Ohta et al., 2013), and the MLEE corpus (Pyysalo et al., 2012). Several techniques have been employed to tackle the problem, ranging from rule-based to machine learning based systems (Vanegas et al., 2015). Recently, neural methods have shown to provide state-of-the-art performance on the task, using either Convolutional Neural Networks (CNNs) (Björne and Salakoski, 2018) or Long Short-Term Memory (LSTM) networks (Li et al., 2019). However, due to the lack of explicit data for evaluating edge detection, most work only report end-to-end performance of EE systems. Further, biomedical corpora comprise many textual variations. According to the language variety space proposed by Plank (2016), each corpus could be characterized by several factors, including the *topic*, the *genre*, and the *language* used, amongst others. Adapting trained models to different language varieties would be desirable to enable cross-domain generalizability.

Although domain adaptation has received increasing importance in other fields, little and scattered work has been done so far in biomedical EE. Vlachos and Craven (2012) showed that a simple supervised domain adaptation approach (Daumé, 2007) is beneficial in handling the differences between abstracts and full-texts, i.e., what we hereafter refer to as *textual scope*, in GE11. However, their work assumed labeled data is available in the target domain, and that the textual scope is the only source of language variation. Nguyen and Grishman (2015) conducted experiments in the newswire domain, showing that CNNs without any external features are more robust than other statistical approaches for the trigger detection stage. Miwa and Ananiadou (2015) integrated weighting and covariate shift into their EE system showing how these methods could improve recall at the cost of precision, while Miwa et al. (2013) proposed a multi-corpus learning approach combining semantic annotations shared across corpora, heuristically filtering corpus-specific annotation instances. Although these works are the closest to our goal, data and performance evaluation results for edge detection in isolation are not available. Further, since our goal is to enable robust cross-domain generalization of models for edge detection on unseen and unannotated domains, we expressly avoid to filter likely spurious negative edge instances based on the knowledge of instances in other corpora (see Section 3.2.2).

3. Data

In this section we present the corpora used in this study, the commonalities and differences in their language aspects, as well as how we use them to generate standardized data for edge detection to enable cross-domain experimentation.

3.1. Corpora and Linguistic Variations

We focus on five biomedical corpora annotated for event structures. These corpora are GE11, ID11, EPI11, PC13, and MLEE. While the first four originate from Shared Tasks, MLEE results from an independent effort towards the annotation of events at various levels of the biological organization. All the corpora share the *genre* and the *language* aspects, since they all derive from scientific publications in English taken from PubMed[2] and PMC[3].

[2] https://www.ncbi.nlm.nih.gov/pubmed/
[3] https://www.ncbi.nlm.nih.gov/pmc/

	linguistic variations		number of documents		
corpus	**textual scope**	**sub-domain**	**set**	**abstracts**	**full-texts**
GE11	full-texts*	reactions about transcription factors in human blood cells	*train*	800	5
			dev	150	5
			test	260	4
ID11	full-texts	mechanisms of infectious diseases in 2-component regulatory systems	*train*	–	15
			dev	–	5
			test	–	10
EPI11	abstracts only	epigenetics and post-translational modifications	*train*	600	–
			dev	200	–
			test	400	–
PC13	abstracts only	reactions about some pathway models in BioModels and PantherDB	*train*	260	–
			dev	90	–
			test	175	–
MLEE	abstracts only	angiogenesis (formation of new blood vessels from pre-existing ones)	*train*	131	–
			dev	44	–
			test	87	–

Table 1: The linguistic aspects (or variations) and the number of documents in the biomedical EE corpora. *The corpus, in addition to full-texts, also contains abstracts from other documents.

Despite these commonalities, the corpora differ along many other language aspects. The main aspects – or linguistic variations – we examine are the *sub-domain* and the *textual scope* (Table 1). The sub-domain is the subject topic the corpus belongs to. It has been previously shown that different sub-domains exhibit different vocabulary, syntax, as well as discourse and sentential features (Lippincott et al., 2011). The sub-domain is a fuzzy aspect, since documents could span different topics with various degrees, and it is implicitly induced in the data collection step of the corpus creation. For example, all the five corpora under consideration are sampled according to different research questions, resulting in different sub-domains. Further, the textual scope of the documents in the corpora introduces another important variation, since abstracts and full-texts noticeably differ in content and structure (Cohen et al., 2010). While EPI11, PC13, and MLEE consist of abstracts only, GE11 and ID11 comprise full-text documents. Note that the presence of a full-text document implies the presence of its corresponding abstract too.

These differences, reported in Table 1 along with the number of documents of each dataset, show that there is no corpus that shares more than one language aspect with another one. The language variation among corpora provides the motivation for conducting a thorough cross-domain study for edge detection.

3.2. Data for Edge Detection

Candidate edge examples are required in order to train and evaluate an edge detector. However, since the standard evaluation in biomedical EE cares about complete event structures only, edges are not explicitly evaluated, and test data annotations in Shared Tasks corpora are blind. Thus, except for the MLEE corpus, we can only use the *train* and *dev* portions (see Table 1) to the purpose. Moreover, there are multiple ways one can generate negative examples, leading to incomparability issues among individual efforts (Pyysalo et al., 2008). In this section we outline how we dealt with this problem, creating standardized benchmark data from all five corpora. As corpora are stand-off annotations, we provide unified preprocessing, i.e., we devise the extraction of edges from event structures and the mapping to unified edge types (Section 3.2.1) and the generation of negative examples (Section 3.2.2). In order to allow for the extension to future corpora and the creation of a wide-coverage system, we propose to focus on the most widely used edge types across all corpora (Section 3.2.3).

3.2.1. Preprocessing of Event Structures

Each document in a corpus is accompanied by two annotation files, one for entities and one for both triggers and event structures. Since an edge is a subset of an event and its endpoints could be both triggers and entities, edges are implicitly encoded in both annotation files. We thus used these files in order to divide event structures into a set of intra-sentence edge examples.[4] We use the scispaCy model with custom postprocessing rules for sentence segmentation (Neumann et al., 2019). Similarly to Miwa et al. (2013), we also handle name variations on the labels that refer to the same edge type,[5] mapping them to their canonical type (e.g., $\{Theme, Theme2, Theme3\} \mapsto Theme$). Due to both the differences in the topic of texts – thus, in the provided edge annotations – and the goal of a cross-domain study, we retain all the semantic edge types which are annotated in multiple corpora. These edge types are `Theme`, `Cause`, `Site`, `CSite`,[6] `AtLoc`, `ToLoc`, and `FromLoc`. For the grouping of these edges refer to Section 3.2.3 while for the formal definition of the edge types refer to the original publications of corpora.

3.2.2. Generation of Negative Examples

For each sentence in the corpus, we generate edge pairs from each trigger to each of its potential arguments (i.e.,

[4] Recent work show an high number of false positives on systems dealing with inter-sentence edges (Lever and Jones, 2016).

[5] In the corpora these are used to arbitrarily enumerate multiple arguments of the same type starting from the same trigger.

[6] `CSite` is the equivalent of `Site` for `Cause` arguments.

			Theme		Cause		Location		NoEdge	
corpus	**set**	**edges**	**#**	**%**	**#**	**%**	**#**	**%**	**#**	**%**
GE11	*train/dev*	28,718	9,027	31.43%	1,082	3.77%	487	1.70%	18,122	63.10%
	test	9,083	2,905	31.98%	442	4.87%	181	1.99%	5,555	61.16%
ID11	*train/dev*	6,430	1,270	19.75%	212	3.30%	28	0.44%	4,920	76.52%
	test	2,805	453	16.15%	113	4.03%	21	0.75%	2,218	79.07%
EPI11	*train/dev*	4,410	1,578	35.78%	145	3.29%	582	13.20%	2,105	47.73%
	test	1,502	518	34.49%	53	3.53%	188	12.52%	743	49.47%
PC13	*train/dev*	24,327	4,958	20.38%	1,834	7.54%	286	1.18%	17,249	70.90%
	test	8,809	1,782	20.23%	635	7.21%	98	1.11%	6,294	71.45%
MLEE	*train/dev*	19,903	3,482	17.49%	1,001	5.03%	219	1.10%	15,201	76.38%
	test	9,415	1,688	17.93%	466	4.95%	91	0.97%	7,170	76.16%

Table 2: Statistics of edges in all the corpora in the newly created *training/development* and *test* sets.

triggers or entities). Similarly to previous work (Björne and Salakoski, 2015), we limit the generation of candidate edges to valid edges only, as defined in the guidelines of each corpus. This yields candidate pairs that are useful for learning, and avoids a highly unbalanced distribution of negative examples with respect to positive examples.[7] Then, each candidate edge which does not have a gold annotated type (e.g., `Theme`, `Cause`, etc.) is labeled as a `NoEdge` type (i.e., a negative instance). In the case an edge type is not among the overlapping edge types in the corpora (Section 3.2.1), we discard the instance. As a result, we obtain a dataset of candidate edges for each corpus that can be used for training and testing.

3.2.3. Merging of Under-Represented Classes

Some classes are highly under-represented. For instance, in the ID11 training set, there is only one instance (0.02%) for both `AtLoc` and `ToLoc` edges, and there are no `AtLoc` instances at all in the dev set. In the same corpus, no `CSite` instances are present in the training set, while one instance (0.04%) is present in the dev set. In the MLEE training and dev sets, there are only 8 instances (0.04%) of `FromLoc` edges, and 5 (0.05%) in the test set. In general, `Site`, `CSite`, `AtLoc`, `ToLoc`, and `FromLoc` are minority classes which are difficult to learn due to the few number of examples. While for `Site` the problem is less pronounced, accounting on average for 3.16% of the edges among corpora, other edges are more problematic. On average, in the training sets there are 0.13% of `CSite`, 0.26% of `AtLoc`, 0.15% of `ToLoc`, and 0.08% of `FromLoc` instances. Since all these edges encode a similar semantic meaning of location, we created a new `Location` class, mapping them into it (i.e., $\{Site, CSite, AtLoc, ToLoc, FromLoc\} \mapsto Location$). This strategy overcomes the learning issues from under-represented classes and provides a mean for cross-domain experimentation, since all corpora now have a `Location` type.

3.2.4. Edge Statistics Across Corpora

The final statistics of the edges in all the corpora are presented in Table 2. The instances in the *train/dev* set are the ones generated from the original training set of the respective corpus, while the *test* instances are the ones coming from the original development set, since no event annotations are provided in the test sets. For the MLEE corpus, where test set annotations are available, the *train/dev* and the *test* sets reflect the original splits of the corpus.

[7] For instance, a `Binding` trigger cannot have a `Phosphorylation` as a `Theme` edge, thus the pair `(Trigger:Binding, Argument:Phosphorylation)` is not produced.

4. Experiments

In this section we present the baseline model used in our experiments, the experimental setup, including the tuning of the hyper-parameters, and an ablation study to investigate the importance of different input embeddings.

4.1. Model Overview

We cast the edge detection problem as a multi-class classification problem where the labels to be predicted are `Theme`, `Cause`, `Location`, and `NoEdge`. We employ a CNN architecture as our framework, following its recent success in biomedical EE (Björne and Salakoski, 2018). The neural network is composed of an input layer, a convolutional layer, a max-pooling layer, and a classification layer. To introduce a non-linearity, we use the ReLU activation function at each layer, except the output layer, which uses softmax. Given a sentence S containing a candidate edge, an example is modelled as its sequence of tokens $\{t_i, ..., t_n\} \in S$. Each token t_i is turned into a real-valued, vectorial representation x_i representing its different syntactic and semantic characteristics. This token-wise representation is the result of the concatenation of different embeddings:

- **Word embedding**: a vectorial representation for the token from pre-trained word embeddings resulting from millions of PubMed abstracts, PMC full-texts, and English Wikipedia texts (Pyysalo et al., 2013). Out-of-vocabulary tokens are randomly initialized;
- **Position embedding**: a vector encoding the relative position of the current token from each target (Zeng et al., 2014). Since the targets are two – the source and the target of the edge to guess – two embeddings are used, one for the source and one for the target;
- **Type embedding**: a vector for the trigger type (or the named entity type) associated with the token, available in the gold annotations of the corpora;

- **POS embedding**: a vector for the POS (*Part-Of-Speech*) tag the token is assigned. We predict POS tags using a biomedical model trained on GENIA 1.0 Treebank and OntoNotes 5.0 (Neumann et al., 2019);
- **Dependency embedding**: we use the path embeddings by Björne and Salakoski (2018), encoding the shortest undirected dependency path from each token to the source and target tokens of the edge candidate. We set the path depth to 2, since a depth $d > 2$ has been reported to hurt the performance in edge detection. As a result, we employ a total of four embeddings (one for the source and one for the target, both at a path distance 1 and 2). Similarly to our POS embeddings, we use a model trained on biomedical texts to predict dependency trees (Neumann et al., 2019).

The sentence representation is thus passed through the convolutional layer and the max-pooling layer. The 4-class classification is done at the classification layer using softmax. Similarly to Nguyen and Grishman (2015) we used shuffled mini-batches of size 50 during training and a dropout regularization rate $\rho = 0.5$ to avoid overfitting. All the weights of the network are updated at training time, except for the 200-dimensional pre-trained word embeddings.

4.2. Experimental Setup

Before training, we tuned the hyper-parameters of the network under a 5-fold *stratified group* cross-validation setting on the *train/dev* set of GE11 (Table 2).[8] We designed this multifaceted cross validation setting (i) to account for the class imbalance, ensuring different splits have examples from all the classes, especially the under-represented ones, and (ii) to avoid the same document falling into different splits, a long-standing issue in comparability of relation extraction systems (Pyysalo et al., 2008), which we extend to the document scope. In fact, not only the same sentence but contiguous sentences could share common information that could lead to an overestimation of performance.

We collect hyper-parameter choices that have been employed in related work (Nguyen and Grishman, 2015; Björne and Salakoski, 2018) for the optimizer, the learning rate, the batch size, the window size, and the number of filters. Additionally, we search for the optimal dimension of the input embeddings, which we concatenate to the 200-dimensional word embeddings. We perform a grid search to select the values, averaging the performance of models for each combination of input embeddings across the five executions. To prevent overfitting, we use early stopping with a patience value of 5 epochs, choosing models from the epoch with the highest micro F_1 score on the development set. Table 3 depicts the search space, where we highlight the best hyper-parameter values we choose. We also find that no significant differences in performance are given by different dimensionalities for each input embedding.

Finally, we train the network on the whole *train/dev* set of each corpus, evaluating it on the respective *test* set (Table 2). For cross-domain evaluation, we instead test all the models – trained on a source *train/dev* corpus – on the *test* set of all the other corpora.

To give a detailed picture of the performance, we report both the micro F_1 and macro F_1 scores, while we use micro F_1 for model selection. We believe reporting both is useful to the community since the evaluation is typically specific to the use case – in fact, one could be more interested in good performance among all classes rather than at an instance level. Additionally, for each metric we also provide the scores considering negative instances in the evaluation (i.e., with `NoEdge`) and without considering them (i.e., without `NoEdge`). We believe this sheds light on the impact of negative examples in edge detection. For the sake of comparability of future results, we also make the *stratified group* splits for each dataset publicly available.

[8] We use GE11 since it represents the largest corpus and it includes both abstract and full-text documents.

Parameter	Search space	Best value
Optimizer	{Adadelta, Adagrad, Adam Adamax, RMSProp, SGD}	Adam
Learning rate	{1., .1, .01, .001, .0005, .0001}	.0005
Batch size	{50, 64}	50
Window sizes	{[3,4,5], [1,3,5,7]}	[3,4,5]
Filters	{32, 150}	150
Emb. size	{8, 16, 32, 50, 64}	32

Table 3: The search space for the best hyper-parameter configuration, along with the optimal values.

Model	Micro F_1 score	Difference
All the input embeddings	**88.83** $\pm$ 0.35	
– POS	88.74 $\pm$ 0.58	-0.09
– TYP	87.15 $\pm$ 0.66	-1.68
– DEP	86.87 $\pm$ 0.33	-1.96
– POS, TYP	86.76 $\pm$ 0.52	-2.07
– POS, DEP	86.67 $\pm$ 0.51	-2.16
– TYP, DEP	84.97 $\pm$ 0.61	-3.86
– POS, TYP, DEP	84.55 $\pm$ 0.57	-4.28

Table 4: The ablation study about input embeddings. We report mean and standard deviation of micro F_1 scores on the development splits for each variant of the model, as well as the performance loss with respect to the complete model.

4.3. Ablation Study

We investigate the contribution of different combinations of input embeddings (i.e., *POS*: part-of-speech, *TYP*: type, *DEP*: dependency) to the performance of the model on the GE11 development splits (Table 4). We average micro F_1 scores of each variant of the model on the five development splits, also reporting the standard deviation. This experiment shows that (i) the most informative input embedding is DEP, which individually contributes 1.96 F_1, followed by TYP, which contributes 1.68 F_1; (ii) POS is the least informative embedding since whether it is removed individually or with other embeddings it decreases the performance only slightly (from 0.09 to 0.42 F_1); (iii) the behaviour of the different input embeddings is consistent both individually and in group, with POS and DEP being highly independent from TYP due to their semantic interdependency; (iv) the inclusion of all the embeddings contributes to a gain of

	target → / source ↓	micro F_1						macro F_1					
		GE11	ID11	EPI11	PC13	MLEE	Avg	GE11	ID11	EPI11	PC13	MLEE	Avg
with NoEdge	GE11	**88.65**	86.67	84.35	84.36	84.79	*-3.71*	**81.01**	**74.28**	77.09	53.28	50.85	*-17.13*
	ID11	80.48	**90.05**	71.50	81.84	84.07	*-10.58*	56.90	65.76	52.33	49.61	48.17	*-14.01*
	EPI11	73.67	78.00	**87.88**	76.17	71.41	*-13.07*	62.22	54.93	**82.19**	48.95	42.78	*-29.97*
	PC13	83.87	86.95	73.10	**88.11**	87.15	*-5.34*	56.81	54.38	54.57	**77.48**	56.00	*-22.04*
	MLEE	81.22	88.20	70.24	84.54	**90.20**	*-9.15*	55.54	55.42	52.54	57.75	**74.59**	*-19.28*
without NoEdge	GE11	**83.66**	69.31	83.43	66.14	63.39	*-13.09*	**77.47**	**68.39**	74.35	40.66	37.37	*-22.28*
	ID11	70.67	**72.63**	59.72	61.54	61.82	*-9.19*	47.17	56.20	42.99	36.33	33.96	*-16.09*
	EPI11	63.69	53.01	**87.17**	52.18	47.93	*-32.97*	56.23	44.53	**80.06**	36.70	29.76	*-38.25*
	PC13	74.92	68.09	59.90	**76.22**	70.08	*-7.97*	46.01	41.70	44.51	**72.42**	43.79	*-28.42*
	MLEE	69.15	67.68	53.48	64.75	**75.95**	*-12.08*	44.92	42.79	42.58	46.67	**68.08**	*-23.84*

Table 5: Cross-domain performance of the baseline model for edge detection. Different performance views are presented, according to both evaluation metric used (i.e., micro F_1 score or macro F_1 score, on the columns) and whether the scores consider the classification of negative examples (i.e., with `NoEdge` or without `NoEdge`, on the rows). In-domain results are on the diagonals (with a *grey* background), while best results on target corpora are in bold. For each combination of metric and evaluation strategy, we indicate the average out-of-domain drop (in italic).

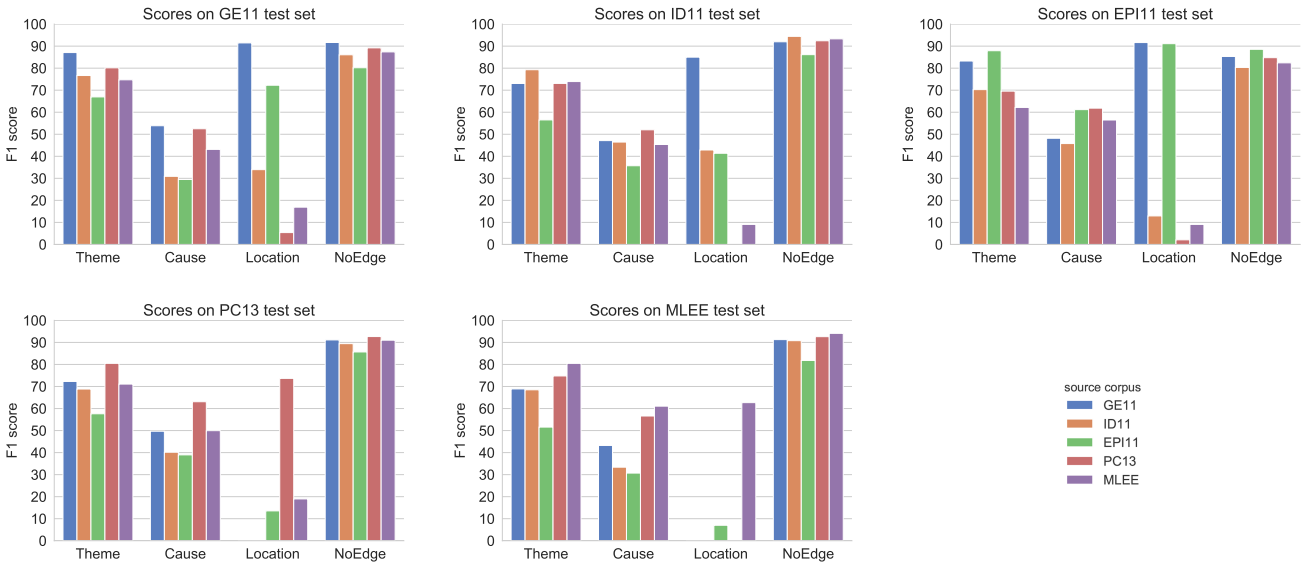

Figure 2: Performance of in-domain models for each *source* corpus on detecting and classifying edge labels on all the other corpora (*target*). Each plot indicates a *target* corpus the *source* model is tested.

4.28 F_1 points with respect to the baseline with only word and position embeddings. It is worth noting that DEP is the most informative embedding despite being a predicted feature. To sum up, the semantic and syntactic features together help edge detection, with the dependency path being the most informative feature.

5. Results and Discussion

In-domain and out-of-domain results of the baseline model across the five corpora are reported in Table 5. Particularly, we present four evaluation strategies to guide the reader in choosing the most appropriate approach for assessing edge detection performance according to its use case. To give additional insights on the results, we also present per-class F_1 scores of each model trained on a *source* corpus when applied to each *target* corpus (Figure 2). Thus, we hereafter use both views to complement the discussion of the results.

In-Domain Results As we can see in Table 5, regardless of the metric and the classes used, in-domain results (with a *grey* background) are consistently better than out-of-domain results. This is not surprising since corpora are characterized by important linguistic variations. The only exception is the ID11 corpus: a model trained on ID11 seems not enough to provide the highest macro F_1 score on the ID11 test set. This is due to both the relatively small size of the ID11 *train/dev* set and the very few training examples having `Location` as a label, clearly insufficient to learn the patterns that characterize the class (Table 2). This only impacts the macro F_1 score, where under-represented classes such as `Location` are given the same weight as dominant classes such as `Theme` and `NoEdge`. As a matter of fact, the GE11 model achieves a higher macro F_1 score on the ID11 test set only because of the `Location` classification performance, almost two times the one provided by the in-domain ID11 model (Figure 2, "Scores on

ID11 test set"). On average over all five corpora, the in-domain micro F_1 is 88.98 and 79.13 with and without considering `NoEdge`, respectively, while the in-domain macro F_1 is 76.21 and 70.85 with and without negative instances, respectively.

Out-of-Domain Results A large drop in performance occurs when in-domain models are applied on *out-of-domain* corpora. As reported in Table 5, the drop in micro F_1 score is from 3.71 to 13.07 points if we consider negative instances in the evaluation, and from 7.97 to 32.97 without considering them. Regarding the macro F_1 score, the drop is even more pronounced, going from 14.01 to 29.97 considering negative edges, and from 16.09 to 38.25 without them. From a closer point of view, we notice EPI11 is the most difficult domain a model could be applied to, as shown by the highest drop in out-of-domain performance across all metrics and classes (i.e., -13.07 and -29.97 considering the `NoEdge` class, and -32.97 and -38.25 without considering the `NoEdge` class, for micro and macro F_1 scores, respectively). This could be due to how EPI11 was constructed, since it is the only corpus that was built avoiding a sample selection bias towards particular proteins or event expressions (Ohta et al., 2011). Another interesting finding is about the ability of some in-domain models to generalize reasonably well to a specific target domain. This is the case of the GE11 model, when applied to EPI11 as a target (i.e., GE11→EPI11), and of the PC13 model, when applied to MLEE (i.e., PC13→MLEE). Although they are far from the performance of the target in-domain models – especially under the macro F_1 metric – they consistently achieve better results than other in-domain models on all metrics and classes. As we can see in Figure 2, "Scores on EPI11 test set", in the GE11→EPI11 case the GE11 model obtains lower performance mainly due to the classification of the `Cause` class, while maintaining close performance on `Location` and `NoEdge` classes. Regarding PC13→MLEE, the difference in performance with respect to the MLEE in-domain model could be explained by the `Location` score, which is 0% (Figure 2, "Scores on MLEE test set").

`Location` seems to be the trickiest class to predict especially in the MLEE target test set, where the only source that achieves a score greater that 0% is EPI11 (7.02%) (Figure 2, "Scores on MLEE test set"). In general, our experiments highlight that there is no single source corpus in which a model could be trained to robustly and consistently achieve good performance on all target corpora. This highlights the urgent need of domain adaptation techniques to make in-domain models able to generalize across linguistic varieties, even within biomedicine itself.

Metrics and Classes We notice important distinctions when using micro F_1 or macro F_1 score as the evaluation metric. Firstly, the scores using macro F_1 are generally lower than the scores using micro F_1. This is because over-represented classes (e.g., `Theme`, `NoEdge`) dominate the micro F_1 score, while the correct classification of under-represented classes (e.g., `Location`) is central to obtain a high macro F_1 score. Secondly, considering negative instances (i.e., `NoEdge`) in computing the averaged scores of edge detection leads to an over-estimation of the performance. This could be explained by the fact that `NoEdge` is the majority class in all the corpora, thus giving a high contribution on both micro and macro F_1 scores. Despite it is a common practice to consider only true annotated labels (i.e., `Theme`, `Cause`, and `Location`) in the evaluation – using wrongly predicted negative instances as *false negatives* for the actual class, and treating as *false positives* for the actual class the instances that are negatives, but classified in that class – we believe considering the negative class in the evaluation could be beneficial in developing real world applications, where a high recall is crucial. Whatever evaluation strategy is used, we see the trend of the scores is consistent across metrics and classes.

Domain and Annotation Adaptation While training on the union of all data is a common domain adaptation baseline (Miwa et al., 2013), it assumes supervision. In contrast, this work encourages research efforts where the target domain is assumed to be unknown and unlabeled, a more difficult but more realistic scenario. We thus plan to explore unsupervised domain adaptation in future work. Due to the lack of a body of research on domain adaptation from data with non-overlapping labels, in this work we assume source and target domains having the same set of edge types. However, we highly value the need of what we call *annotation adaptation* in the near future, i.e., the adaptation of a model from partially overlapping source and target labels.

6. Conclusions

We provided the first cross-domain evaluation study for biomedical edge detection, together with standardized data from five gold-standard corpora to enable further progress in comparable edge detection. We proposed different evaluation strategies to assess the performance of models, together with an in-domain baseline for edge detection, for which we assessed the contribution of different combinations of input embeddings, finding syntactic and semantic features to be particularly helpful. We used in-domain models to assess the performance drop across five datasets, shedding light on the importance of developing robust models that could deal with the linguistic variations in different corpora. We believe this work could encourage future work in domain adaptation, and could sensitize an awareness about the language differences within the biomedical domain. The data, the splits, and the baselines for edge detection are publicly available at `https://www.cosbi.eu/cfx/9985`.

7. Acknowledgements

We thank Rob van der Goot and Marija Stepanovic for the valuable comments that improved the manuscript.

8. Bibliographical References

Ananiadou, S., Pyysalo, S., Tsujii, J., and Kell, D. B. (2010). Event extraction for systems biology by text mining the literature. *Trends in Biotechnology*, 28(7):381–390.

Björne, J. and Salakoski, T. (2015). TEES 2.2: Biomedical Event Extraction for Diverse Corpora. *BMC Bioinformatics*, 16(16):S4.

Björne, J. and Salakoski, T. (2018). Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. In *Proceedings of the BioNLP 2018 workshop (ACL 2018)*, pages 98–108.

Cohen, K. B., Johnson, H. L., Verspoor, K., Roeder, C., and Hunter, L. E. (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, 11(492).

Daumé, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263.

Huang, C. C. and Lu, Z. (2016). Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings in Bioinformatics*, 17(1):23–32.

Kim, J.-d., Wang, Y., Tagagi, T., and Yonezawa, A. (2011). Overview of Genia Event Task in BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop (ACL 2011)*, pages 7–15.

Lever, J. and Jones, S. J. (2016). VERSE: Event and Relation Extraction in the BioNLP 2016 Shared Task. In *Proceedings of the 4th BioNLP Shared Task Workshop (ACL 2016)*, pages 42–49.

Li, D., Huang, L., Ji, H., and Han, J. (2019). Biomedical Event Extraction based on Knowledge-driven Tree-LSTM. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1421–1430.

Lippincott, T., Séaghdha, D. T., and Korhonen, A. (2011). Exploring subdomain variation in biomedical language. *BMC Bioinformatics*, 12(212).

Miwa, M. and Ananiadou, S. (2015). Adaptable, high recall, event extraction system with minimal configuration. *BMC Bioinformatics*, 16(10).

Miwa, M., Pyysalo, S., Ohta, T., and Ananiadou, S. (2013). Wide coverage biomedical event extraction using multiple partially overlapping corpora. *BMC Bioinformatics*, 14(1).

Neumann, M., King, D., Beltagy, I., and Ammar, W. (2019). ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327.

Nguyen, T. H. and Grishman, R. (2015). Event Detection and Domain Adaptation with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 365–371.

Ohta, T., Pyysalo, S., and Tsujii, J. (2011). Overview of the epigenetics and post-translational modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop (ACL 2011)*, pages 16–25.

Ohta, T., Pyysalo, S., Rak, R., Rowley, A., Chun, H.-W., Jung, S.-J., Jeong, C.-H., Choi, S.-P., Tsujii, J., and Ananiadou, S. (2013). Overview of the Pathway Curation (PC) task of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop (ACL 2013)*, pages 67–75.

Plank, B. (2016). What to do about non-standard (or non-canonical) language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*.

Pyysalo, S., Sætre, R., Tsujii, J., and Salakoski, T. (2008). Why biomedical relation extraction results are incomparable and what to do about it. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 149–152.

Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., Sobral, B., Tsujii, J., and Ananiadou, S. (2011). Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop (ACL 2011)*, pages 26–35.

Pyysalo, S., Ohta, T., Miwa, M., Cho, H. C., Tsujii, J., and Ananiadou, S. (2012). Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):575–581.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional Semantics Resources for Biomedical Text Processing. *Proceedings of the 5th Languages in Biology and Medicine Conference (LBM 2013)*, pages 39–44.

Vanegas, J. A., Matos, S., González, F., and Oliveira, J. L. (2015). An Overview of Biomolecular Event Extraction from Scientific Documents. *Computational and mathematical methods in medicine*, 2015.

Vlachos, A. and Craven, M. (2012). Biomedical event extraction from abstracts and full papers using search-based structured prediction. *BMC Bioinformatics*, 13:1–11.

Walker, C., Strassel, S., Medero, J., and Maeda, K. (2006). ACE 2005 multilingual training corpus.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 2335–2344.