Learning to Extract Biological Event and Relation Graphs

Jari Björne[1], **Filip Ginter**[1], **Juho Heimonen**[2], **Sampo Pyysalo**[3] and **Tapio Salakoski**[1,2]
[1]Department of IT, University of Turku
[2]Turku Centre for Computer Science (TUCS)
Joukahaisenkatu 3-5, 20520 Turku, Finland
firstname.lastname@utu.fi
[3]Department of Computer Science, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-0033 Tokyo, Japan
smp@is.s.u-tokyo.ac.jp

Abstract

While the overwhelming majority of information extraction efforts in the biomedical domain have focused on the extraction of simple binary interactions between named entity pairs, some recently published corpora provide complex, nested and typed event annotations that aim to accurately capture the diversity of biological relationships. We present the first machine learning approach for extracting such relationships, utilizing both a graph kernel and a novel, task-specific feature set. We show that relationships can be predicted with 77% F-score, or 83% if their type and direction is disregarded. Using both gold standard and generated parses, we determine the impact of parsing on extraction performance. Finally, we convert our predicted complex relationships to binary interactions, recovering binary annotation with 62% F-score, relating the new method to the large body of work available on binary interactions.

1 Introduction

The previous decade has brought about an ever-increasing interest in the application of natural language processing methods to address information overload challenges in the biomedical domain (see, e.g., the recent review by Zweigenbaum et al. (2007)). Most domain information extraction (IE) efforts have focused on relationships between biologically interesting molecules. Among these, the most prominent IE target are protein-protein interactions (PPIs). The overwhelming majority of proposed approaches cast the task as determining which pairs of co-occurring entities are related (binary interactions). Many methods further specify the nature of these relationships by assigning them types or specifying the roles (e.g. agent/patient) that the entities play. While this extraction model has supported considerable advances in biomedical IE and has served as the basis for real-world applications for e.g. assisted database curation (Alex et al., 2008), its limitations, such as the restriction to events between entity pairs commonly referred to as binary interactions in the domain literature, are increasingly recognized by the biomedical NLP community. In this paper, we argue for an alternate model and present the first machine-learning approach to the extraction of structured, complex events and relationships among bioentities.

To overcome the limitations of the pairwise approach to biomedical IE, two recent corpora, BioInfer (Pyysalo et al., 2007a) and the GENIA Event corpus (Kim et al., 2008a) annotate events and static relationships using a more expressive formalism that differs from the prevailing approach in several key aspects: First, type, direction and the trigger statement in the text stating the relationship (often a verb) are annotated. Second, events can have more than two participants whose roles are specified, allowing the accurate representation of statements such as *proteins A, B and C form a complex*. Finally, events can also act as arguments of other events, enabling the annotation of nested events such as *A causes B to bind C* (Figure 1A). These representations largely resemble *event extraction* as formulated in (later) Message Understanding Conferences (MUC) (see, e.g., Sundheim (1995)) and in the Automatic Content Extraction (ACE) program (see, e.g., Doddington (2004)). BioInfer also annotates static relations (e.g. *substructure*) and both BioInfer and GENIA annotate non-biological relationships (e.g. coreferences) with specialized mechanisms. In this paper, we use the term *complex relationship* to encompass both event and generic relationship annotation.

Kristiina Jokinen and Eckhard Bick (Eds.)
NODALIDA 2009 Conference Proceedings, pp. 18–25

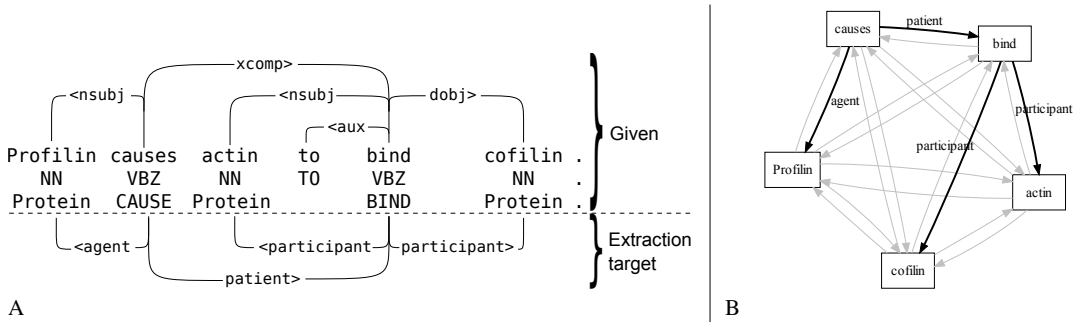

Figure 1: **A.** An example sentence that shows the dependency parse and the relationship graph, whose edges we aim to predict. **B.** Relationship edges can exist between any of the annotated entities and events. For each pair, there can be one undirected or two directed relationships.

In this paper we first introduce the corpora used and their conversion to examples usable for machine learning, then the criteria used for evaluating the system followed by our results. The distinct task of binarization is discussed in its own section. Finally we provide an overview of the related work in this field followed by conclusions.

2 Methods

2.1 Corpora and the Extraction Task

BioInfer consists of 1100 sentences with both semantic and syntactic annotation. For GENIA, we use the 1968 sentence intersection of the GENIA Treebank (syntactic annotation) and GENIA Event corpus (semantic annotation). For developing our system, we used half of each corpus. The other half alone was used for the final experiments to avoid overfitting our system to the data.

In order to use the two corpora for IE, their annotations have to be cast in a single, consistent representation (Figure 1A). Here we follow Björne et al. (2008) and Heimonen et al. (2008) in representing the semantic annotations as graphs whose nodes correspond to entities and events, and labeled directed edges to their relationships. The relationship edges describe *themes* and *causes* of events, structural relations between physical entities such as *substructure* and also non-biological relations such as coreferences. These graphs capture the several distinct forms of annotation in the corpora in a unified, yet expressive format.

The corpora are further processed for our IE task (Figure 2). All entities and events must be represented by a trigger in the text, a constraint imposed to assure that they can be recognized using regular text tagging methods. Some event nodes, like the semantic equality in *actin A (ActA)* that

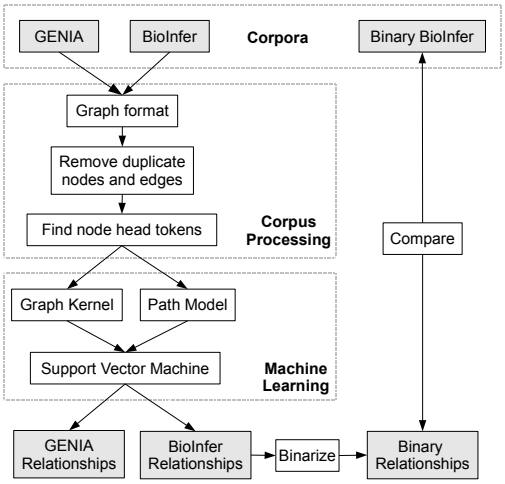

Figure 2: Outline of the experiments. Corpora are converted to a shared graph representation from which the edges are learned. Binarization of predicted BioInfer relationships allows comparison with a binary version of the corpus.

defines a relationship between *actin A* and *ActA* do not have an explicit trigger word. This type of node and its participant edges are collapsed into an equivalent relationship edge.

Dependency representations of syntax are commonly applied in IE. We use both hand-annotated gold-standard data provided with the corpora as well as parses generated using the Charniak-Lease parser (Lease and Charniak, 2005), which is one of the best-performing parsers in the biomedical domain, achieving an F-score of 81.3% on GENIA and 79.4% on BioInfer (Pyysalo et al., 2007b). All parses are transformed to the Stanford dependency scheme using the tools of de Marneffe et al. (2006). As illustrated in Figure 1A, the dependencies of the parse form a graph that often closely

resembles the relationship graph. Roughly 60% of BioInfer and GENIA relationship edges correspond to a single dependency (Björne et al., 2008).

While the nodes of the dependency graph are tokens, the nodes of the relationship graph are entities and events whose triggers can span multiple tokens. To align the graphs, the trigger of each entity or event is associated with one token, its semantic head. This mapping produces a textbound semantic graph representation (*relationship graph*) that is largely equivalent in information content to the original corpus annotations.

We note that multiple entities or events can occasionally have the same trigger. Since IE systems start from a trigger, producing multiple events or entities of the same type is a non-trivial task which is outside the scope of this study. We represent these cases with one node in the relationship graph. Especially in the case of events, this can lead to some loss of information. In situations like *A and B bind C and D, respectively*, there are two distinct events with the same trigger *bind*.

To summarize, we cast our IE task as one of generating the edges of the relationship graph (Figure 1B) given its nodes, i.e. events and entities. Here we follow the standard division of IE research into identification of entities and subsequent extraction of their relationships, focusing on the subtask of relationship extraction. This definition was chosen as it most resembles the related task of extracting binary protein-protein interactions, which can be viewed as a special case of relationship edges. This allows the straightforward application of already existing methods.

Note that both GENIA and BioInfer only annotate events with explicitly stated participants. Therefore an event with no participants in the relevant span of text (a sentence in BioInfer and a document in GENIA) are not annotated and thus will not be considered for potential relationships.

We perform two main information extraction experiments. First, we extract *untyped undirected* relationships, i.e. detect whether a pair of nodes has a relationship of any type or direction. Second, we extract *typed directed* relationships, where we determine if two nodes have a relationship, in which direction it is defined, and what its type is.

2.2 Defining examples

If a single pair has several relationships of the same direction but different types , these would result in identical examples. To be able to use standard classifiers that give one classification per example, we merge the types of such examples into one compound type. As seen in Tables 2 and 3 this is extremely rare. We define one example per pair per direction for the typed directed task and one example per pair for the untyped undirected task (Figure 1B). Pairs with an annotated relationship are the positive examples and, as per the closed world assumption, those with no relationship are the negative examples.

For machine learning, each example is represented as a set of features. We compare two feature generation methods (Figure 2). The graph kernel was chosen as we represent the complex relationships in a graph format. For an overview of this recent state-of-the-art method and its use in the extraction of binary interactions we refer to Airola et al. (2008) and Miwa et al. (2008). Since the graph kernel has high memory and processing time requirements, we also developed a new, smaller feature set specifically targeting complex relationships.

2.3 Path Model

The Path Model feature set was developed to be highly specific for the extraction of complex relationships. For each pair of nodes, a number of features are generated. Most of these are based on the shortest path in the syntactic dependency graph (Figure 1A). While the graph kernel uses weights to emphasize tokens and dependencies on the shortest path, our path model aims to capture their relations explicitly.

The shortest path is defined as the shortest undirected path in the dependency graph that connects the head tokens of the two nodes (entities/events) of the example pair. Since multiple paths can exist between tokens in the Stanford dependency scheme, there can be several shortest paths. In such cases, all of them are used to generate features. If no path exists, only the head tokens of the node pair are used for generating features.

Most features are built from the attributes of the tokens and dependencies of the parse. For tokens, these attributes include the text of the token, the part of speech tag (using the Penn Treebank tagset) and the entity/event type (such as *protein* for an entity or *bind* for an event). If the token belongs to a named entity (e.g. a known protein name like *actin*) its text is replaced with a generic place-

holder to prevent the system from making predictions based on the frequency of relationships between specific names. The attributes of a dependency are its type (e.g. *subject*) and direction relative to its surrounding dependencies. Unless otherwise stated, all features are binary, that is, they have a value of 1 or 0 (present/absent).

***N*-grams** For each shortest path, a number of *n*-grams are generated by merging the attributes of 2-4 consecutive tokens. Similarly, *n*-grams are built from the types and directions of consecutive dependencies. For each token (resp. dependency), an additional 3-gram merging its attributes with the attributes of its two flanking dependencies (resp. tokens) is defined. Finally, a 2-gram is defined for each pair of consecutive tokens, arranged in the order of their governor-dependent relationship. All of these *n*-grams aim to explicitly state the structural relations that the graph kernel defines only indirectly.

Hanging Dependency Features Tokens immediately outside the path connected by dependencies to the terminal tokens of the path contain information about the context of the two nodes of the example pair. These dependencies "hanging" at the ends of the path are used to define features, as are the tokens they link to.

Individual Component Features For all of the tokens and dependencies on the shortest paths, features are also defined based only on their attributes in isolation of their context. Tokens within the triggers of the two nodes of the example pair are tagged to explicitly state this role. Additional features are defined for each token stating its position at either the terminus or the interior of the path.

Frequency Features The number of tokens in the shortest path is defined as the value of the *length*-feature, as well as explicitly as a *length_n* feature. The number of occurrences of each entity/event type (such as *protein* or *bind*) in the sentence are defined as values of specific features.

Relationship Graph Node Features For the two nodes of each example, features are defined from the combination of their categories (*entity* or *event*) as well as their types (such as *protein* or *bind*). If the triggers of both nodes have the same head token, a feature is defined explicitly representing this potential self-loop.

2.4 Machine Learning

For classification, we use the support vector machine as implemented in SVM^{light} (for the untyped undirected task) and SVM^{multiclass} (for the typed directed task) by Joachims (1999). All experiments are performed using ten-fold cross-validation. Examples are divided into ten sets on the basis of articles, avoiding the information leak between training and testing described by Sætre et al. (2007). For each of the ten folds, the classifier is trained on the union of eight of the sets. One set is used for a grid search for the optimal SVM regularization parameter C and the remaining set is the test set, separating parameter selection from testing.

2.5 Evaluation Criteria

We use two measures to evaluate our results: the standard F-score metric (the harmonic mean of precision and recall) and AUC.

F-score is a common metric for evaluating relationship extraction, but is sensitive to the class distribution of the data. For binary classification (untyped undirected relationships), the true/false positives/negatives from which F-score is calculated are easily defined. For multiclass classification (typed directed relationships), we have a negative class (i.e. no relationship) and a number of positive classes (the relationship types). F-scores are micro-averaged to take into account the number of instances in each class. For the micro-average, correctly classified non-negative examples are true positives, examples incorrectly classified as instances of a non-negative class are false positives and non-negative examples incorrectly classified as negatives are false negatives.

AUC, or area under the receiver operating characteristic curve, is a class distribution invariant binary performance measure (Hanley and McNeil, 1982). This and other advantages have led to AUC becoming widely adopted in machine learning.

3 Results and Discussion

The performance of the feature generation methods for both the untyped undirected and the typed directed tasks is shown in Table 1. Performance on both tasks is well above the trivial all-positive baseline. For the untyped undirected task, detecting the presence of an edge has the highest F-score of 83% on BioInfer with gold standard parses. As expected, F-score is lower with parses generated

			untyped undirected				typed directed		
corpus	parse	features	P	R	F	AUC	P	R	F
BioInfer	GS	PM	84.4	82.1	83.1±2.3	89.4±1.8	78.7	76.7	77.7±2.6
		GK	74.9	70.6	72.6±2.6	82.6±2.2	72.6	56.8	63.6±2.5
	CL	PM	76.6	67.3	71.5±4.6	81.4±2.6	73.5	61.9	67.0±3.7
		GK	66.8	61.4	63.8±2.4	77.3±1.5	64.2	47.1	54.1±4.1
GENIA	GS	PM	75.5	63.1	68.7±1.5	80.5±1.2	70.2	60.9	65.2±2.4
	CL	PM	72.3	57.4	63.8±2.8	77.6±2.1	65.6	55.5	60.1±3.0

Table 1: Performance of relationship extraction using gold standard (GS) and Charniak-Lease (CL) parses. Examples are classified based on either the path model (PM) or features produced by the graph-kernel (GK). (P)recision, (R)ecall, (F)-score and AUC are shown with standard deviations for F and AUC. For the *typed directed* task, all scores are micro-averaged. The all-positive baseline F-score for the *untyped undirected* task is 31% for BioInfer and 17.1% for GENIA.

by the Charniak-Lease parser (71% on BioInfer), showing the extent to which the parser limits extraction performance.

The path model outperforms the graph kernel for both untyped undirected and typed directed extraction. Despite weighting the shortest path, the graph kernel produces features from the entire sentence for each example, thus resulting in a large number of potentially misleading features. The graph kernel also lacks all explicit *n*-grams of the path model. Due to its excessive computational requirements, we only apply the graph kernel to the smaller BioInfer dataset.

Predicting types and directions turns the problem into a multi-class classification task. The micro-averages in Table 1 show that this does not notably decrease performance. Compared to the untyped undirected task, F-scores are 3-6 percentage points lower with the path model and 9-10 percentage points lower with the graph kernel. This relatively small difference is promising for future work, as type and direction are important for defining meaningful complex relationships.

Information extraction performance for individual BioInfer relationship edge types is shown in Table 2. Promisingly the most important group for defining biologically interesting relationships, the event-group, shows high precision and recall for all of its types. Many static relationships, e.g. edges of type *identity*, *possessor* and *sub* (we refer to Heimonen (2008) for definitions) can be extracted with even higher reliability, perhaps due in part to a close correspondence to specific syntactic structures, such as prepositional phrases. On the other hand, edges representing complex syntactic structures, such as coreferences (*corefer*) are recovered with lower accuracy, as can be expected since coreference resolution is best addressed using a specialized method. Merged edges are a result of having one edge per pair of nodes per direction (see Section 2.2). These very rare cases are not recovered by the learning-based approach.

group	type	count	P	R	F
event	participant	836	80.0	77.2	78.6
	patient	655	79.7	77.4	78.5
	agent	428	75.5	66.8	70.9
static	identity	289	86.5	88.9	87.7
	sub	134	85.5	79.1	82.2
	possessor	119	83.2	83.2	83.2
	member	105	64.8	43.8	52.3
	super	59	78.2	72.9	75.4
	nesting	20	66.7	50.0	57.1
non-biol.	equal	120	60.5	60.0	60.3
	corefer	66	55.6	22.7	32.3
	rel-ent	22	0.0	0.0	0.0
merged	contain+sub	20	0.0	0.0	0.0
	member+agent	3	0.0	0.0	0.0
	agent+patient	3	0.0	0.0	0.0
	f-contain+sub	2	0.0	0.0	0.0

Table 2: Per-type results of extraction of *typed directed* relationships from BioInfer using gold standard parses and the path model. Count shows the number of examples of a given type from a total of 31674 including negatives.

Performance per GENIA edge type is shown in Table 3. Non-biological relationships, such as coreferences, are syntactically diverse structures and have unsurprisingly a low performance. *Cause* and *theme* types define the participants of events and roughly correspond to the *agent* and *patient* types of BioInfer, respectively. The *participant* type of BioInfer describes relationships that can be thought of as either *agent* or *patient*. GENIA uses the *theme* type for such cases.

The high performance for both BioInfer and GENIA typed directed relationship extraction is especially noticeable in light of the very high class imbalance. Even for the most common types the

group	type	count	P	R	F
event	theme	3164	73.6	65.1	69.1
	cause	1202	65.3	54.7	59.5
non-biol.	coref	252	51.2	25.4	34.0
	scatter	169	40.0	17.8	24.6
merged	cause+theme	1	0.0	0.0	0.0

Table 3: Per-type results of extraction of *typed directed* relationships from GENIA using gold standard parses and the path model. Count shows the number of examples of a given type from a total of 104198 including negatives.

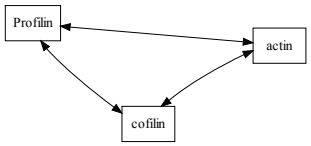

Figure 3: Untyped undirected binary relationships. Compare with Figure 1B. In this example, all possible binary relationships exist.

positive/negative ratio is about 0.03. The most common BioInfer type, *participant*, has 836 positives vs. 30838 negatives (Table 2). For GENIA, the most common type, *theme*, has 3164 positives vs. 101034 negatives (Table 3).

We tested the impact of the feature groups defined in Section 2.3 by disabling one group at a time. F-score decreased at most less than 2 percentage points, indicating a substantial overlap of information between the groups. We also tried defining the features without entity/event types, which reduced F-score by 4.4 percentage points, indicating that this information is important but not critical for the system.

4 Binarization

The prevailing approach in the domain is to extract binary interactions, that is, relationships restricted to occurring between pairs of physical entities (most often proteins). To compare the performance of the proposed approach to these existing extraction systems, the semantically rich relationship graphs must be reduced into a less-expressive, binarized form. Examples of binary relationships are shown in Figure 3.

The transformation from a complex to a binary relationship graph has been shown to be possible for BioInfer (Heimonen et al., 2008). This binarization process aims to express as binary relationships the biologically relevant information present in complex relationships, while minimizing the inevitable loss of information. Consider, for example, the sentence *Phosphorylation of cofilin regulates actin polymerization*, which expresses the events *regulation*, *phosphorylation* and *polymerization* among the proteins *cofilin* and *actin*. It can be summarized with a binary relationship *regulation* while the information regarding *phosphorylation* and *polymerization* is lost.

corpus	parse	P	R	F
BioInfer	GS	74.2	53.7	62.3
	CL	70.7	42.9	53.4

Table 4: Performance of binary relationship extraction measured against the binarized gold standard BioInfer relationship annotation for which the F-score of the all-positive baseline is 40.8%.

The predicted typed directed complex relationship graphs for BioInfer were binarized using the software of Heimonen et al. (2008). The output was evaluated against the binarized gold standard BioInfer relationship annotation. To compare with previously published results on this dataset, we treat the relationships as untyped undirected. The results of the evaluation are presented in Table 4. The F-score of 53.4% for the Charniak-Lease parsed data should be related to the F-score of 61.3% reported by Airola et al. (2008). This difference can be partly explained by the fact that the binarizer was developed for hand-annotated data rather than noisy, automatically generated data. Also, the precision of 70.7% suggests that complex relationships recovered by the system to the point that they could be binarized were often correct. We have thus shown that the output of an IE system targeting complex relationship graphs can be binarized, although this process currently results in lower performance than extraction methods directly targeting binary interactions.

5 Related Work

Extraction of protein relationships is a key task in biomedical NLP, and has been widely studied in the simple setting of recognizing pairs of related co-occurring entities. The problem has been considered in recent shared tasks (Nedéllec, 2005; Krallinger et al., 2008) as well as in dozens of studies employing a variety of different corpora for training and evaluation (Pyysalo et al., 2008).

Several recently proposed extraction methods

make use of dependency representations of syntax (Kim et al., 2008b; Miwa et al., 2008), including the Stanford dependency representation (Airola et al., 2008; Van Landeghem et al., 2008; Katrenko and Adriaans, 2008). Many of the features we apply are standard in relation extraction studies; for a recent study of "ACE-style" feature sets see the study by Buyko et al. (2008).

By contrast to the wealth of IE studies focusing on pairs of related entities, has received much less attention. While hand-written systems capable of extracting structured events (Friedman et al., 2001) have been proposed, the present study is to the best of our knowledge the first to consider the task of learning to extract events as represented in the BioInfer and GENIA corpora. Further, while task settings similar to ours have been widely considered in the MUC and ACE evaluations and part of the task setting shares many characteristics with semantic role labeling as considered e.g. in the recent CoNLL evaluation (Surdeanu et al., 2008), meaningful comparison across domains and resources would be difficult to establish. In relating our results to those of previously proposed methods, we will thus only consider biomedical relationship extraction results as they relate to our results for binarized relation extraction.

Due to the difficulty of meaningful comparison of reported results across different corpora (Airola et al., 2008; Van Landeghem et al., 2008), we will consider our results in comparison with recently proposed methods evaluated on the AIMed corpus (Bunescu et al., 2005), which is frequently used in domain studies (Bunescu et al., 2005; Giuliano et al., 2006; Airola et al., 2008; Van Landeghem et al., 2008; Miyao et al., 2008; Miwa et al., 2008) and can be seen as an emerging *de facto* standard for biomedical relationship extraction method evaluation. Among these comparable studies, the best results are reported by Miwa et al. (2008) using the graph kernel of Airola et al. (2008), considered also in the present study. We note that Airola et al. (2008) report an F-score of 61% on the BioInfer corpus for the binary relationship extraction task. Given that our method is not primarily intended for this type of binary PPI extraction and that our binarization method was not originally developed to deal with noisy input, we find our result of 53% F-score on BioInfer (62% with gold standard parses) encouraging.

The system described in this paper formed the basis for the best-performing system in the primary task of the BioNLP'09 Shared Task on Event Extraction,[1] further validating the presented approach and results (Björne et al., 2009).

6 Conclusions

We provide the first system designed for extracting complex relationships as defined in the BioInfer and GENIA Event corpora, using the complex semantic annotation they provide that allows interaction extraction between a broader set of biological concepts than only named molecules. The unified graph format abstracts from the various information extraction tasks and defines a shared representation for the layers of annotation in both BioInfer and the GENIA Event corpus. This abstraction provides a representation approachable for the general NLP community lacking extensive knowledge of the biological details.

Classification performance of the system, even on typed and directed data, was good, and having a system that predicts typed events (e.g. binding or phosphorylation) provides valuable data when extracting specific information about a defined biological issue. By binarizing our predicted relationship graphs, we have shown that complex relationship extraction need not be a completely separate problem from binary interaction extraction.

As a contribution to the emerging field of complex relationship extraction, we will publish the software used to convert GENIA and BioInfer to the shared graph format, the extraction system and the software used for binarizing the extracted complex relationships.

Acknowledgments

This work was funded by the Academy of Finland. We thank CSC - IT Center for Science Ltd. for providing computational resources.

References

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2.

B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, S. Roebuck, R. Tobin, and X. Wang. 2008. Assisted curation: Does text mining really help? In *Proc. of PSB'08*.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask

J. Björne, S. Pyysalo, F. Ginter, and T. Salakoski. 2008. How complex are complex protein-protein interactions? In *Proc. of SMBM'08*, pages 125–128.

J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski. 2009. Extracting complex biological events with rich graph-based features sets. In *Proc. of the BioNLP'09 Workshop at NAACL-HLT 2009*. To appear.

R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani, and Y. Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

E. Buyko, E. Beisswanger, and U. Hahn. 2008. Testing different ACE-style feature sets for the extraction of gene regulation relations from MEDLINE abstracts. In *Proc. of SMBM'08*, pages 21–28.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. The Automatic Content Extraction (ACE) program: Tasks, data, and evaluation. In *Proc. of LREC'04*, pages 837–840.

C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. 2001. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(Suppl. 1):S74–S82.

C. Giuliano, A. Lavelli, and L. Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proc. of EACL'06*, pages 401–408.

J. A. Hanley and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

J. Heimonen, S. Pyysalo, F. Ginter, and T. Salakoski. 2008. Complex-to-pairwise mapping of biological relationships using a semantic network representation. In *Proc. of SMBM'08*.

T. Joachims, 1999. *Advances in Kernel Methods - Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press.

S. Katrenko and P. Adriaans. 2008. A local alignment kernel in the context of nlp. In *Proc. of Coling'08*.

J-D. Kim, T. Ohta, and Tsujii J. 2008a. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):10.

S. Kim, J. Yoon, and J. Yang. 2008b. Kernel approaches for genic interaction extraction. *Bioinformatics*, 24(1):118–126.

M. Krallinger, F. Leitner, C. Rodriguez-Penagos, and A. Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(Suppl 2):S4.

M. Lease and E. Charniak. 2005. Parsing biomedical literature. In *Proc. of the Second International Joint Conference on Natural Language Processing*, Lecture notes in computer science, pages 58–69.

M.-C. de Marneffe, B. MacCartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC'06*, pages 449–454.

M. Miwa, R. Sætre, Y. Miyao, T. Ohta, and J. Tsujii. 2008. Combining multiple layers of syntactic information for protein-protein interaction extraction. In *Proc. of SMBM'08*.

Y. Miyao, R. Sætre, K. Sagae, T. Matsuzaki, and J. Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proc. of ACL'08*, pages 46–54.

C. Nedéllec. 2005. Learning Language in Logic – genic interaction extraction challenge. In *Proc. of the 4th ICML Workshop on Learning Language in Logic*, pages 31–37, Aug.

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. 2007a. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50.

S. Pyysalo, F. Ginter, V. Laippala, K. Haverinen, J. Heimonen, and T. Salakoski. 2007b. On the unification of syntactic annotations under the stanford dependency scheme: A case study on BioInfer and GENIA. In *Proc. of BioNLP'07*, pages 25–32.

S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6.

R. Sætre, K. Sagae, and J. Tsujii. 2007. Syntactic features for protein-protein interaction extraction. In *Second International Symposium on Languages in Biology and Medicine short papers*.

B. M. Sundheim. 1995. Overview of results of the MUC-6 evaluation. In *Proc. of MUC-6*, pages 13–31.

M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of CoNLL'08*, pages 159–177.

S. Van Landeghem, Y. Saeys, B. De Baets, and Y. Van de Peer. 2008. Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *Proc. of SMBM'08*.

P. Zweigenbaum, D. Demner-Fushman, H. Yu, and K. B. Cohen. 2007. Frontiers of biomedical text mining: current progress. *Briefings in Bioinformatics*, 8(5):358–375.

ISSN 1736-6305 Vol. 4
http://hdl.handle.net/10062/9206