Adapting a General Semantic Interpretation Approach to Biological Event Extraction

Halil Kilicoglu and Sabine Bergler
Department of Computer Science and Software Engineering
Concordia University
1455 de Maisonneuve Blvd. West
Montréal, Canada
{h_kilico,bergler}@cse.concordia.ca

Abstract

The second BioNLP Shared Task on Event Extraction (BioNLP-ST'11) follows up the previous shared task competition with a focus on *generalization* with respect to text types, event types and subject domains. In this spirit, we re-engineered and extended our event extraction system, emphasizing linguistic generalizations and avoiding domain-, event type- or text type-specific optimizations. Similar to our earlier system, syntactic dependencies form the basis of our approach. However, diverging from that system's more pragmatic nature, we more clearly distinguish the shared task concerns from a general semantic composition scheme, that is based on the notion of *embedding*. We apply our methodology to core bio-event extraction and speculation/negation detection tasks in three main tracks. Our results demonstrate that such a general approach is viable and pinpoint some of its shortcomings.

1 Introduction

In the past two years, largely due to the availability of GENIA event corpus (Kim et al., 2008) and the resulting shared task competition (BioNLP'09 Shared Task on Event Extraction (Kim et al., 2009)), event extraction in biological domain has been attracting greater attention. One of the criticisms towards this paradigm of corpus annotation/competition has been that they are concerned with narrow domains and specific representations, and that they may not generalize well. For instance, GENIA event corpus contains only Medline abstracts on transcription factors in human blood cells. Whether models trained on this corpus would perform well on full-text articles or on text focusing on other aspects of biomedicine (e.g., treatment or etiology of disease) remains largely unclear. Since annotated corpora are not available for every conceivable domain, it is desirable for automatic event extraction systems to be generally applicable to different types of text and domains without requiring much training data or customization.

	GENIA	**EPI**	**ID**	**BB**	**BI**
# core events	9	15	10	2	10
Triggers?	Y	Y	Y	N	N
Full-text?	Y	N	Y	N	N
Spec/Neg?	Y	Y	Y	N	N

Table 1: An overview of BioNLP-ST'11 tracks

In the follow-up event to BioNLP'09 Shared Task on Event Extraction, organizers of the second BioNLP Shared Task on Event Extraction (BioNLP-ST'11) (Kim et al., 2011a) address this challenge to some extent. The theme of BioNLP-ST'11 is *generalization* and the net is cast much wider. There are 4 event extraction tracks: in addition to the GENIA track that again focuses on transcription factors (Kim et al., 2011b), the epigenetics and post-translational modification track (EPI) focuses on events relating to epigenetic change, such as DNA methylation and histone modification, as well as other common post-translational protein modifications (Ohta et al., 2011), whereas the infectious diseases track (ID) focuses on bio-molecular mechanisms of infectious diseases (Pyysalo et al., 2011a). Both GENIA and ID tracks include data pertaining to full-text articles, as well. The fourth track, Bacteria, consists of two sub-tracks: Biotopes (BB) and Interactions (BI) (Bossy et al. (2011) and Jourde

Proceedings of BioNLP Shared Task 2011 Workshop, pages 173–182,
Portland, Oregon, USA, 24 June, 2011.

et al. (2011), respectively). A summary of the BioNLP-ST'11 tracks is given in Table (1).

We participated in three tracks: GENIA, EPI, and ID. In the spirit of the competition, our aim was to demonstrate a methodology that was general and required little, if any, customization or training for individual tracks. For this purpose, we used a two-phase approach: a syntax-driven *composition* phase that exploits linguistic generalizations to create a general semantic representation in a bottom-up manner and a *mapping* phase, which relies on the shared task event definitions and constraints to map relevant parts of this semantic representation to event instances. The composition phase takes as its input simple entities and syntactic dependency relations and is intended to be fully general. On the other hand, the second phase is more task-specific even though the kind of task-specific knowledge it requires is largely limited to event definitions and trigger expressions. In addition to extracting core biological events, our system also addresses speculation and negation detection within the same framework. Our results demonstrate the feasibility of a methodology that uses little training data or customization.

2 Methodology

In our general research, we are working towards a linguistically-grounded, bottom-up discourse interpretation scheme. In particular, we focus on lower level discourse phenomena, such as *causation*, *modality*, and *negation*, and investigate how they interact with each other, as well as their effect on basic propositional semantic content (who did what to who?) and higher discourse/pragmatics structure. In our model, we distinguish three layers of propositions: *atomic, embedding*, and *discourse*. An *atomic proposition* corresponds to the basic unit and lowest level of meaning: in other words, a semantic relation whose arguments correspond to ontologically simple entities. Atomic propositions form the basis for *embedding propositions*, that is, propositions taking as arguments other propositions (embedding them). In turn, embedding and atomic propositions act as arguments for *discourse relations*[1]. Our main motivation in casting the problem of discourse interpretation in this structural manner is two-fold: a) to explore the semantics of the embedding layer in a systematic way b) to allow a bottom-up semantic composition approach, which works its way from atomic propositions towards discourse relations in creating general semantic representations.

The first phase of our event extraction system (*composition*) is essentially an implementation of this semantic composition approach. Before delving into further details regarding our implementation for the shared task, however, it is necessary to briefly explain the embedding proposition categorization that our interpretation scheme is based on. With this categorization, our goal is to make explicit the kind of semantic information expressed at the embedding layer. We distinguish three basic classes of embedding propositions: MODAL, ATTRIBUTIVE, and RELATIONAL. We provide a brief summary below.

2.1 MODAL type

The embedding propositions of MODAL type *modify* the status of the embedded proposition with respect to its factuality, possibility, or necessity, and so on. They typically involve a) judgement about the status of the proposition, b) evidence for the proposition, c) ability or willingness, and d) obligations and permissions, corresponding roughly to EPISTEMIC, EVIDENTIAL, DYNAMIC and DEONTIC types (cf. Palmer (1986)), respectively. Further subdivisions are given in Figure (1). In the shared task context, the MODAL class is mostly relevant to the speculation and negation detection tasks.

2.2 ATTRIBUTIVE type

The ATTRIBUTIVE type of embedding serves to *specify* an attribute of an embedded proposition (semantic role of an argument). They typically involve a verbal predicate (*undergo* in Example (1) below), which takes a nominalized predicate (*degradation*) as one of its syntactic arguments. The other syntactic argument of the verbal predicate corresponds to a semantic argument of the embedded predicate. In Example (1), *p105* is a semantic argument of PATIENT type for the proposition indicated by *degradation*.

[1]Discourse relations, also referred to as *coherence* or *rhetorical relations* (Mann and Thompson, 1988), are not relevant to the shared task and, thus, we will not discuss them further in this paper.

(1) . . . *p105 undergoes* ***degradation*** . . .

Verbs functioning in this way are plenty (e.g., *perform* for the AGENT role, *experience* for *experiencer* role). With respect to the shared task, we found that the usefulness of the ATTRIBUTIVE type of embedding was largely limited to verbal predicates *involve* and *require* and their nominal forms.

2.3 RELATIONAL type

The RELATIONAL type of embedding serves to semantically *link* two propositions, providing a discourse/pragmatic function. It is characterized by permeation of a limited set of discourse relations to the clausal level, often signalled lexically by "discourse verbs" (Danlos, 2006) (e.g., *cause*, *mediate*, *lead*, *correlate*), their nominal forms or other abstract nouns, such as *role*. We categorize the RELATIONAL class into CAUSAL, TEMPORAL, CORRELATIVE, COMPARATIVE, and SALIENCY types. In the example below, the verbal predicate *leads to* indicates a CAUSAL relation between the propositions whose predicates are highlighted.

(2) ***Stimulation*** *of cells leads to a rapid* ***phosphorylation*** *of I*κ*B*α . . .

While not all the subtypes of this class were relevant to the shared task, we found that CAUSAL, CORRELATIVE, and SALIENCY subtypes play a role, particularly in complex regulatory events. The portions of the classification that pertain to the shared task are given in Figure (1).

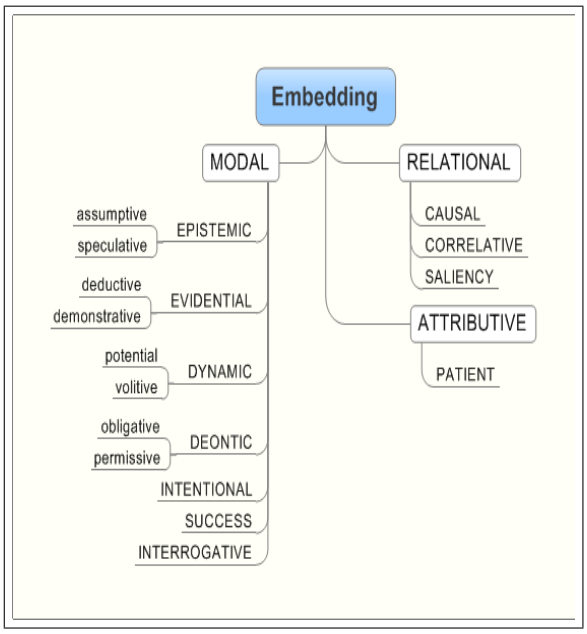

Figure 1: Embedding proposition categorization relevant to the shared task

3 Implementation

In the shared task setting, embedding propositions correspond to complex regulatory events (e.g., Regulation, Catalysis) as well as event modifications (Negation and Speculation), whereas atomic propositions correspond to simple event types (e.g., Phosphorylation). While the treatment of these two types differ in significant ways, they both require that simple entities are recognized, syntactic dependencies are identified and a dictionary of trigger expressions is available. We first briefly explain the construction of the trigger dictionary.

3.1 Dictionary of Trigger Expressions

In the previous shared task, we relied on training data and simple statistical measures to identify good trigger expressions for events and used a list of triggers that we manually compiled for speculation and negation detection (see Kilicoglu and Bergler (2009) for details). With respect to atomic propositions, our method of constructing a dictionary of trigger expressions remains essentially the same, including the use of statistical measures to distinguish good triggers. The only change we made was to consider affixal negation and set polarity of several atomic proposition triggers to *negative* (e.g., *nonexpression*, *unglycosylated*). On the other hand, we have been extending our manually compiled list of speculation/negation triggers to include other types of embedding triggers and to encode finer grained distinctions in terms of their categorization and trigger behaviors. The training data provided for the shared task also helped us expand this trigger dictionary, particularly with respect to RELATIONAL trigger expressions. It is worth noting that we used the same embedding trigger dictionary for all three tracks that we participated in. Several entries from the embedding trigger dictionary are summarized in Table (2).

Lexical polarity and *strength* values play a role in the composition phase in associating a context-dependent scalar value with propositions. Lexical polarity values are largely derived from a polarity lexicon (Wilson et al., 2005) and extended by us-

Trigger	POS	Semantic Type	Lexical Polarity	Strength
show	VB	DEMONSTRATIVE	*positive*	1.0
unknown	JJ	EPISTEMIC	*negative*	0.7
induce	VB	CAUSAL	*positive*	1.0
fail	VB	SUCCESS	*negative*	0.0
effect	NN	CAUSAL	*neutral*	0.5
weakly	RB	HEDGE	*neutral*	-
absence	NN	REVERSE	*negative*	-

Table 2: Several entries from the embedding dictionary

ing heuristics involving the event types associated with the trigger[2]. Some polarity values were assigned manually. Some strength values were based on prior work (Kilicoglu and Bergler, 2008), others were manually assigned. As Table (2) shows, in some cases, the semantic type (e.g., DEMONSTRATIVE, CAUSAL) is simply a mapping to the embedding categorization. In other cases, such as *weakly* or *absence*, the semantic type identifies the role that the trigger plays in the composition phase. The embedding trigger dictionary incorporates ambiguity; however, for the shared task, we limit ourselves to *one semantic type per trigger* to avoid the issue of disambiguation. For ambiguous triggers extracted from the training data, the semantic type with the maximum likelihood is used. On the other hand, we determined the semantic type to use manually for triggers that we compiled independent of the training data. In this way, we use 466 triggers for atomic propositions and 908 for embedding ones[3].

3.2 Composition

As mentioned above, the composition phase assumes simple entities, syntactic dependency relations and trigger expressions. Using these elements, we construct a semantic embedding graph of the document. To obtain syntactic dependency relations, we segment documents into sentences, parse them using the re-ranking parser of Charniak and Johnson (2005) adapted to the biomedical domain (McClosky and Charniak, 2008) and extract syntactic dependencies from parse trees using the Stanford dependency scheme (de Marneffe et al., 2006). In addition to syntactic dependencies, we also require information regarding individual tokens, including lemma, part-of-speech, and positional information, for which we also rely on Stanford parser tools. We present a high level description of the composition phase below.

3.2.1 From syntactic dependencies to embedding graphs

As the first step in composition, we convert syntactic dependencies into embedding relations. An embedding relation, in our definition, is very similar to a syntactic dependency; it is typed and holds between two textual elements. It diverges from a syntactic dependency in two ways: its elements can be multi-word expressions and it is aimed at better reflecting the direction of the semantic dependency between its elements. Take, for example, the sentence fragment in Example (3a). Syntactic dependencies are given in (3b) and the corresponding embedding relations in (3c). The fact that the adjectival predicate in modifier position (*possible*) semantically embeds its head (*involvement*) is captured with the first embedding relation. The second syntactic dependency already reflects the direction of the semantic dependency between its elements accurately and, thus, is unchanged as an embedding relation.

(3) (a) *... possible involvement of HCMV ...*
(b) *amod*(*involvement,possible*)
prep_of(*involvement,HCMV*)
(c) *amod*(*possible,involvement*)
prep_of(*involvement,HCMV*)

To obtain the embedding relations in a sentence, we apply a series of transformations to its syntactic

[2]For example, if the most likely event type associated with the trigger is Negative_regulation, its polarity is considered negative.

[3]Note, however, that not all embedding propositions (or their triggers) were directly relevant to the shared task.

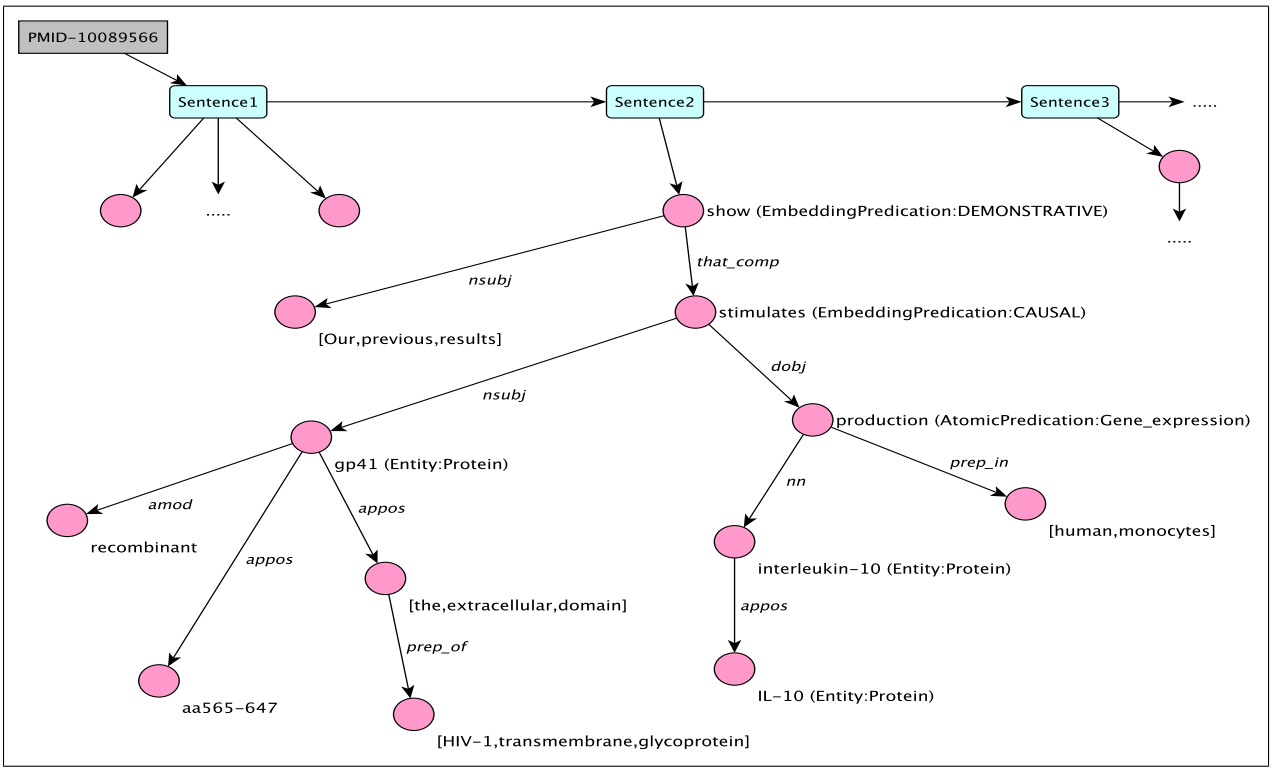

Figure 2: The embedding graph for the sentence *Our previous results show that recombinant gp41 (aa565-647), the extracellular domain of HIV-1 transmembrane glycoprotein, stimulates interleukin-10 (IL-10) production in human monocytes.* in the context of the document embedding graph for the Medline abstract with PMID 10089566.

dependencies. A transformation may not be necessary, as with the *prep_of* dependency in the example above. It may result in collapsing several syntactic dependencies into one, as well, or in splitting one into several embedding relations. In addition to capturing semantic dependency behavior explicitly, these transformations serve to incorporate semantic information (entities and triggers) into the embedding structure and to correct syntactic dependencies that are systemically misidentified, such as those that involve modifier coordination.

After these transformations, the resulting directed acyclic embedding graph is, in the simplest case, a tree, but more often a forest. An example graph is given in Figure (2). The edges are associated with the embedding relation types, and the nodes with textual elements.

3.2.2 Composing Propositions

After constructing the embedding graph, we traverse it in a bottom-up manner and compose semantic propositions. Before this procedure can take place, though, the embedding graph pertaining to each sentence is further linked to the document embedding graph in a way to reflect the proximity of sentences, as illustrated in Figure (2). This is done to enable discourse interpretation across sentences, including coreference resolution.

Traversal of the embedding structure is guided by *argument identification rules*, which apply to non-leaf nodes in the embedding graph. An argument identification rule is essentially a mapping from *the type of the embedding relation* holding between a parent node and its child node and *part-of-speech* of the parent node to a logical argument type (*logical subject*, *logical object* or *adjunct*). Constraints on and exclusions from a rule can be defined, as shown in Table (3). We currently use about 80 such rules, mostly adapted from our previous shared task system (Kilicoglu and Bergler, 2009).

After all the descendants of a non-leaf node are recursively processed for arguments, a semantic proposition can be composed. We define a semantic proposition as consisting of a trigger, a collection

Relation	Applies to	Argument	Constrained to	Exclusions
prep_on	NN	Object	*influence,impact,effect*	-
agent	VB	Subject	-	-
nsubjpass	VB	Object	-	-
whether_comp	VB	Object	INTERROGATIVE	-
prep_in	NN	Adjunct	-	*effect, role, influence, importance*

Table 3: Several argument identification rules. Note that constraints and exclusions may apply to trigger categories, as well as to lemmas.

of core and adjunct arguments as well as a polarity value and a scalar value. The polarity value can be *positive*, *negative* or *neutral*. The scalar value is in the (0,1) range. Atomic propositions are simply assigned polarity value of *neutral*[4] and the scalar value of 1.0. On the other hand, in the context of embedding propositions, the computation of these values, through which we attempt to capture some of the interactions occurring at the embedding layer, is more involved. For the sentence depicted in Figure (2), the relevant resulting embedding and atomic propositions are given below.

(4) DEMONSTRATIVE(em_1,Trigger=*show*, Object=em_2, Subject=*Our previous results*, Polarity=positive, Value=1.0)

(5) CAUSAL(em_2, Trigger=*stimulates*, Object=ap_1, Subject=*recombinant gp41*, Polarity=positive, Value=1.0)

(6) Gene_expression(ap_1, Trigger= *production*, Object= *interleukin-10*, Adjunct= *human monocytes*, Polarity=neutral, Value=1.0)

The composition phase also deals with coordination of entities and propositions as well as with propagation of arguments at the lower levels.

3.3 Mapping Propositions to Events

The goal of the *mapping* phase is to impose the shared task constraints on the partial interpretation achieved in the previous phase. We achieve this in three steps.

The first step is to map embedding proposition types to event (or event modification) types. We defined constraints that guide this mapping. Some of these mappings are presented in Table (4). In this way, Example (4) is pruned, since embedding propositions of DEMONSTRATIVE type satisfy the constraints only if they have negative polarity, as shown in Table (4).

We then apply constraints concerned with the semantic roles of the participants. For this step, we define a small number of *logical argument/semantic role mappings*. These are similar to argument identification rules, in that the mapping can be constrained to certain event types or event types can be excluded from it. We provide some of these mappings in Table (5). With these mappings, the Object and Subject arguments of the proposition in Example (5) are converted to Theme and Cause semantic roles, respectively.

As the final step, we prune event participants that do not conform to the event definition as well as the propositions whose types could not be mapped to a shared task event type. For example, a Cause participant for a Gene_expression event is pruned, since only Theme participants are relevant for the shared task. Further, a proposition with DEONTIC semantic type is pruned, because it cannot be mapped to a shared task type. The infectious diseases track (ID) event type Process is interesting, because it may take no participants at all, and we deal with this idiosyncrasy at this step, as well. This concludes the progressive transformation of the graph to event and event modification annotations.

4 Results and Discussion

With the two-phase methodology presented above, we participated in three tracks: GENIA (Tasks 1 and 3), ID, and EPI. The official evaluation results we obtained for the GENIA track are presented in Table (6) and the results for the EPI and ID tracks in

[4]Unless affixal negation is involved, in which case the assigned polarity value is *negative*.

Track	Prop. Type	Polarity	Value	Correspond. Event (Modification) Type
GENIA,ID	CAUSAL	neutral	-	Regulation
GENIA,ID,EPI	SUCCESS	negative	-	Negation
EPI	CAUSAL	positive	-	Catalysis
GENIA,ID,EPI	SPECULATIVE	-	> 0.0	Speculation
GENIA,ID,EPI	DEMONSTRATIVE	negative	-	Speculation

Table 4: Several event (and event modification) mappings

Logical Arg.	Semantic Role	Constraint	Exclusion
Object	Theme	-	Process
Subject	Cause	-	-
Subject	Theme	Binding	-
Object	Participant	Process	-
Object	Scope	Speculation, Negation	-

Table 5: Logical argument to semantic role mappings

Event Class	Recall	Precis.	F-score
Localization	39.27	90.36	54.74
Binding	29.33	49.66	36.88
Gene_expression	65.87	86.84	74.91
Transcription	32.18	58.95	41.64
Protein_catabolism	66.67	71.43	68.97
Phosphorylation	75.14	94.56	83.73
EVT-TOTAL	52.67	78.04	62.90
Regulation	33.77	42.48	37.63
Positive_regulation	35.97	47.66	41.00
Negative_regulation	36.43	43.88	39.81
REG-TOTAL	35.72	45.85	40.16
Negation	18.77	44.26	26.36
Speculation	21.10	38.46	27.25
MOD-TOTAL	19.97	40.89	26.83
ALL-TOTAL	43.55	59.58	50.32

Table 6: Official GENIA track results, with *approximate span matching/approximate recursive matching* evaluation criteria

Table (7). With the official evaluation criteria, we were ranked 5th in the GENIA track (5/15), 7th in the EPI track (7/7) and 4th in the ID track (4/7). There were only two submissions for the GENIA speculation/negation task (Task 3) and our results in this task were comparable to those of the other participating group: our system performed slightly better with speculation, and theirs with negation.

Our core module extracts adjunct arguments, using ABNER (Settles, 2005) as its source for additional named entities. We experimented with mapping these arguments to non-core event participants (Site, Contextgene, etc.); however, we did not include them in our official submission, because they seemed to require more work with respect to mapping to shared task specifications. Due to this shortcoming, the performance of our system suffered significantly in the EPI track.

A particularly encouraging outcome for our system is that our results on the GENIA development set versus on the test set were very close (an F-score of 51.03 vs. 50.32), indicating that our general approach avoided overfitting, while capturing the linguistic generalizations, as we intended. We observe similar trends with the other tracks, as well. In the EPI track, development/test F-score results were 29.10 vs. 27.88; while, in the ID track, interestingly, our test set performance was better (39.64 vs. 44.21). We also obtained the highest recall in the ID track, despite the fact that our system typically favors precision. We attribute this somewhat idiosyncratic performance in the ID track partly to the fact that we did not use a track-specific trigger dictionary. Most of the ID track event types are the same as those of GENIA track, which probably led to identification of some ID events with GENIA-only triggers[5].

One of the interesting aspects of the shared task was its inclusion of full-text articles in training and evaluation. Cohen et al. (2010) show that structure and content of biomedical abstracts and article bodies differ markedly and suggest that some of these

[5]This clearly also led to low precision particularly in complex regulatory events.

Track-Eval. Type	Recall	Precis.	F-score
EPI-FULL	20.83	42.14	27.88
EPI-CORE	40.28	76.71	52.83
ID-FULL	49.00	40.27	44.21
ID-CORE	50.77	43.25	46.71

Table 7: Official evaluation results for EPI and ID tracks. Primary evaluation criteria underlined.

differences may pose problems in processing full-text articles. Since one of our goals was to determine the generality of our system across text types, we did not perform any full text-specific optimization. Our results on article bodies are notable: our system had stable performance across text types (in fact, we had a very slight F-score improvement on full-text articles: 50.40 vs. 50.28). This contrasts with the drop of a few points that seems to occur with other well-performing systems. Taking only full-text articles into consideration, we would be ranked 4th in the GENIA track. Furthermore, a preliminary error analysis with full-text articles seems to indicate that parsing-related errors are more prevalent in the full-text article set than in the abstract set, consistent with Cohen et al.'s (2010) findings. At the same time, our results confirm that we were able to abstract away from this complexity to some degree with our approach.

We have a particular interest in speculation and negation detection. Therefore, we examined our results on the GENIA development set with respect to Task 3 more closely. Consistent with our previous shared task results, we determined that the majority of errors were due to misidentified or missed base events (70% of the precision errors and 83% of the recall errors)[6]. Task 3-specific precision errors included cases in which speculation or negation was debatable, as the examples below show. In Example (7a), our system detected a Speculation instance, due to the verbal predicate *suggesting*, which scopes over the event indicated by *role*. In Example (7b), our system detected a Negation instance, due to the nominal predicate *lack*, which scopes over the events indicated by *expression*. Neither were annotated as such in the shared task corpus.

(7) (a) *. . . suggesting a* ***role*** *of these 3' elements in beta-globin gene expression.*
(b) *. . . DT40 B cell lines that lack* ***expression*** *of either PKD1 or PKD3 . . .*

Another class of precision errors was due to argument propagation up the embedding graph. It seems the current algorithm may be too permissive in some cases and a more refined approach to argument propagation may be necessary. In the following example, while *suggest*, an epistemic trigger, does not embed *induction* directly (as shown in (8b)), the intermediate nodes simply propagate the proposition associated with the *induction* node up the graph, leading us to conclude that the proposition triggered by *induction* is speculated, leading to a precision error.

(8) (a) *. . . these findings suggest that PWM is able to initiate an intracytoplasmic signaling cascade and EGR-1* ***induction*** *. . .*
(b) *suggest → able → initiate → induction*

Among the recall errors, some of them were due to shortcomings of the composition algorithm, as it is currently implemented. One recall problem involved the embedding status of and rules concerning copular constructions, which we had not yet addressed. Therefore, we miss the relatively straightforward Speculation instances in the following examples.

(9) (a) *. . . the A3G promoter appears constitutively* ***active****.*
(b) *. . . the precise factors that* ***mediate*** *this induction mechanism remain unknown.*

Similarly, the lack of a trigger expression in our dictionary may cause recall errors. The example below shows an instance where this occurs, in addition to lack of an appropriate argument identification rule:

(10) *mRNA was quantified by real-time PCR for FOXP3 and GATA3* ***expression****.*

Our system also missed an interesting, domain-specific type of negation, in which the minus sign indicates negation of the event that the entity participates in.

(11) *. . . CD14- surface Ag* ***expression*** *. . .*

[6]Even a bigger percentage of speculation/negation-related errors in the EPI and ID tracks were due to the same problem, as the overall accuracy in those tracks is lower.

5 Conclusions and Future Work

We explored a two-phase approach to event extraction, distinguishing general linguistic principles from task-specific aspects, in accordance with the *generalization* theme of the shared task. Our results demonstrate the viability of this approach on both abstracts and article bodies, while also pinpointing some of its shortcomings. For example, our error analysis shows that some aspects of semantic composition algorithm (argument propagation, in particular) requires more refinement. Furthermore, using the same trigger expression dictionary for all tracks seems to have negative effect on the overall performance. The incremental nature of our system development ensures that some of these shortcomings will be addressed in future work.

We participated in three supporting tasks, two of which (Co-reference (CO) and Entity Relations (REL) tasks (Nguyen et al. (2011) and Pyysalo et al. (2011b), respectively) were relevant to the main portion of the shared task; however, due to time constraints, we were not able to fully incorporate these modules into our general framework, with the exception of the co-reference resolution of relative pronouns. Since our goal is to move towards discourse interpretation, we plan to incorporate these modules (inter-sentential co-reference resolution, in particular) into our framework. After applying the lessons we learned in the shared task and fully incorporating these modules, we plan to make our system available to the scientific community.

References

Robert Bossy, Julien Jourde, Philippe Bessières, Marteen van de Guchte, and Claire Nédellec. 2011. BioNLP Shared Task 2011 - Bacteria Biotope. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*, pages 173–180.

K Bretonnel Cohen, Helen L Johnson, Karin Verspoor, Christophe Roeder, and Lawrence E Hunter. 2010. The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, 11:492.

Laurence Danlos. 2006. "Discourse verbs" and discourse periphrastic links. In C Sidner, J Harpur, A Benz, and P Kühnlein, editors, *Second Workshop on Constraints in Discourse (CID06)*, pages 59–65.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 449–454.

Julien Jourde, Alain-Pierre Manine, Philippe Veber, Karën Fort, Robert Bossy, Erick Alphonse, and Philippe Bessières. 2011. BioNLP Shared Task 2011 - Bacteria Gene Interactions and Renaming. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Halil Kilicoglu and Sabine Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9 Suppl 11:s10.

Halil Kilicoglu and Sabine Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. In *Proceedings of Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop*, pages 119–127.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9:10.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop*, pages 1–9.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011a. Overview of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011b. Overview of the Genia Event task in BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Meeting of the Association for Computational Linguistics*, pages 101–104.

Ngan Nguyen, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. Overview of the Protein Coreference task in BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. 2011. Overview of the Epigenetics and Post-translational Modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Frank R Palmer. 1986. *Mood and modality*. Cambridge University Press, Cambridge, UK.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. 2011a. Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Sampo Pyysalo, Tomoko Ohta, and Jun'ichi Tsujii. 2011b. Overview of the Entity Relations (REL) supporting task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Burr Settles. 2005. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 347–354.