Refactoring the Genia Event Extraction Shared Task Toward a General Framework for IE-Driven KB Development

Jin-Dong Kim*,[1], Yue Wang[1], Nicola Colic[2], Seung Han Baek[3], Yong Hwan Kim[3], Min Song[3]
[1] Database Center for Life Science (DBCLS)
[2] University of Zürich
[3] Yonsei University

Abstract

For its fourth organization, the Genia event extraction (GE) shared task is refactored toward a general platform for shared information extraction (IE) tasks, and for an IE-driven knowledge base (KB) system. On the newly implemented shared task platform, the GE task is run as an experimental task. The task and the platform has been tested by two teams who cooperated with the organizers. The paper presents the new shared task system and discusses on the experimental submissions.

1 Introduction

Since its first introduction in 2009 as the task of the first BioNLP Shared Task (BioNLP-ST) organization, the Genia event extraction (GE) task has been one of the most investigated IE tasks (Kim et al., 2009; Kim et al., 2011; Kim et al., 2013). The biggest contribution of BioNLP-ST might be that it introduced fine-grained and highly structured information extraction (IE) tasks to the community of biomedical information extraction (BioIE), when the research in the community was weighted toward extracting binary relations (Krallinger et al., 2007; Lu et al., 2004; Chun et al., 2006). Since then the tasks of BioNLP-ST have motivated and nourished the community to develop a number of biomedical event extraction systems (Björne and Salakoski, 2013; Miwa et al., 2010),

Originally designed as tasks based on intrinsic evaluation, however, the tasks of BioNLP-ST could not be free from criticism on unclarity about their impact on real world application (Caporaso et al., 2008). Also, there was a growing need for generalized resources for shared task organization with which the cost of organizing shared tasks could be substantially reduced. With this motivation, for its 4th organization in 2016, the GE task is completely re-designed and re-implemented as an experimental task with two goals.

Firstly, we aim at establishing a seamless connection from the IE task to knowledge base (KB) construction. It means we assume KB construction as the target application of the GE task. Particularly, we aim at developing a KB about NFκB proteins, which is the subject domain the GE task has focused on. In the end, we hope to be able to deliver an end-user service of the KB, so that people who are interested in NFκB proteins can easily access knowledge about them. Toward this end, we automate the process of populating a KB from the output of the task, and solicit working systems to perform the task.

Secondly, we aim at generalizing the resources of shared task organization. Previous iterations of organization showed that shared task is an effective format to promote development of IE solutions. Shared task organization however requires a lot of effort and expertise. If the resources for shared task organization become generalized and readily available, more shared tasks can be easily organized. To this end, we re-designed and re-implemented the shared task resources which have been developed so far for the GE task.

Due to the complexity of refactoring the whole task, instead of being run as a competition among participants, the GE4 task is organized as an experimental task, experimenting newly implemented features, with involvement of voluntary feedback from participants. Finally, two systems could go through up to their final submissions, thanks to which the newly implemented shared task system could be thoroughly tested. Manual analysis on the submissions shows both achievments and remaining issues, which are discussed in the end of this paper.

*Corresponding author, jdkim@dbcls.rois.ac.jp

Proceedings of the 4th BioNLP Shared Task Workshop, pages 23–31,
Berlin, Germany, August 13, 2016. ©2016 Association for Computational Linguistics

2 Design

2.1 Platform

To achieve the first goal of generalizing the shared task system, *PubAnnotation* (Kim and Wang, 2012) was chosen as the platform. There were several reasons for the choice. Firstly, as a public repository of literature annotation, PubAnnotation provides various ways of submitting and accessing annotation data sets, which are fundamental for shared task organization. Secondly, it features an automatic text alignment function, which provides a reliable solution for aligning annotations collected from different groups. Thirdly, it is a near mature system, which has a growing user base with more than hundred of data sets.

While PubAnnotation provides many useful functions, a shared task organization still requires more functions. Most importantly, automatic evaluation needs to be enabled for efficient development of IE systems. Also, to prevent over-fitting the benchmark data set, often the annotations in the benchmark data set are required to be hidden. Accordingly, the two key features are implemented into PubAnnotation, which are described in following sections.

2.1.1 Comparison of annotations

A shared task organization often features an automatic evaluation of predicted annotations. For generalization, we cast it as general comparison of two different annotation sets. On PubAnnotation, an annotation data set is maintained as a *project*, and each project is maintained by its *maintainer*.

A new feature *annotation comparison* is implemented into PubAnnotation. Using the feature, the maintainer of a project can compare the project against any other project. We call the former a *subject project*, and the latter a *reference project*. A comparison is performed by looking at how many annotations in the reference project can be recovered in the subject project. The comparison is calculated in terms of recall, precision, and f-score, in their standard meaning.

As PubAnnotation represents annotations in three types, *denotations*, *relations*, and *modifications*[1], comparison is also performed for each of the three types. In case the subject and reference projects have different sets of documents, comparison is performed only for the documents found in both projects.

With this feature, any corpus with manual annotation can potentially serve as a shared task: any one can attempt to automatically reproduce the manual annotation, and evaluate the accuracy.

2.1.2 Blind annotations

A new feature *blind annotations* is implemented into PubAnnotation, to enable hiding annotations in a certain project. By blinding annotations of a project, individual annotations become inaccessible. However, the project can still be used for comparison. In this way, the project can still function as a benchmark data set.

2.2 Data sets

Data sets prepared for the GE4 task is grouped into *benchmark data sets* and *supporting data sets*.

2.2.1 Benchmark data sets

For the benchmark data set of the GE4 task, the same set of documents used for the GE3 task are cleaned and used again. However, the separation of the data set into training, development and test sets is slightly changed: the training and development data sets are merged into one set which we call a *reference data set*. Thus the GE4 benchmark data set consists of two sets: the reference data set with 20 full papers and the test data set with 14 full papers. The change in dataset separation and naming is made in order to remove the impression that it is a machine learning task and to encourage development of various approaches.

The annotations in the test data set are "blinded" using the newly implemented feature (see section 2.1.2). Following the tradition of BioNLP-ST to provide protein annotations for the test data set, which will allow participants to spend more time for developing their event extraction system, the test data set is duplicated to make what we call a *test-start data set*. The test-start data set is the same as the test data set except for the fact that it has only protein annotations and the annotations are not blinded. Participant can begin their test first by obtaining a copy of the test-start data set. Then, event annotations produced by their systems can be added to it, which will be compared against the test data set for evaluation. The three benchmark data sets for the GE4 task are illustrated on the top of Figure 1.

[1] http://www.pubannotation.org/docs/annotation-format/

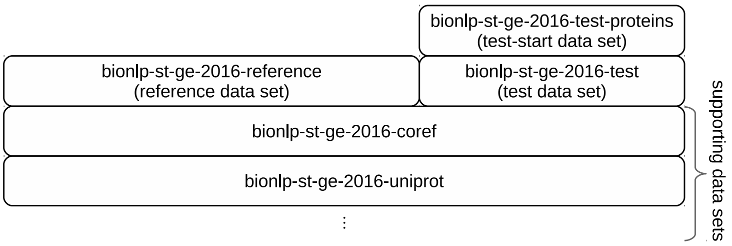


Figure 1: Data sets for the GE4 task

2.2.2 Supporting data sets

Besides the benchmark data sets, other data sets are prepared to support participants to use rich information. Firstly, the coreference annotations from the GE3 data set are separated into an individual annotation set, *bionlp-st-ge-2016-coref*. Secondly, UniProt IDs are annotated to the benchmark data sets, to provide "normalization" or "grounding" of protein annotations. Note that the GE4 task organization aims at constructing an IE-driven KB which requires information pieces to be grounded to database entries. The UniProt ID annotation thus plays an important role in the GE4 task. For the UniProt ID annotation, a simple dictionary matching approach is used, but the dictionary is tailored to the benchmark data sets to raise the accuracy of UniProt ID annotation particularly for the benchmark data sets.

For other supporting data sets, we attempted to collect automatic annotation tools, rather than just collecting static annotation data sets[2]. PubAnnotation has a feature to communicate with web services to obtain annotations, and the feature is used to produce the supporting resources via the automatic annotation tools. It ensures that the same annotations can be produced for new documents. Besides the two sets of annotations described above, two syntactic parsers, and several named entity recognizers are prepared as RESTful web service:

- bionlp-st-ge-2016-uniprot: UniProt ID annotation
- bionlp-st-ge-2016-coref: coreference annotation
- pmc-enju-pas: deep dependency parsing by Enju (Miyao and Tsujii, 2008)
- bionlp-spacy-parsed: dependency parsing spacy (Honnibal et al., 2013)
- UBERON-AE: anatomical entities in UBERON (Mungall et al., 2012)
- ICD10: disease names as defined in ICD10
- GO-BP: biological processes as defined in GO
- GO-CC: cellular components as defined in GO

Note that collection of supporting annotations usually requires a non-trivial effort of organizers, to ensure all the annotations provided by different groups to be precisely aligned to the texts in the benchmark datasets. Otherwise, there is a high chance that the texts may be changed during preprocessing by different groups, which may cause an issue of aligning different versions of texts when they are collected. However, thanks to the automatic alignment algorithm implemented in PubAnnotation (See section 2.1), it is not an issue any more as long as they are collected on PubAnnotation. It is a clear benefit of using PubAnnotation as a platform of shared task organization.

Figure 9 shows excerpts of data sets prepared for the GE4 task. The annotation data sets can be retrieved individually or altogether through the RESTful API. For example, by accessing the following URL, the annotations shown in Figure 9 can be obtained in JSON at once: `http://pubannotation.org/docs/sourcedb/PMC/sourceid/3245220/divs/11/spans/4375-4513/annotations.json?projects=bionlp-st-ge-2016-reference,bionlp-st-ge-2016-uniprot,bionlp-st-ge-2016-coref,pmc-enju-pas,bionlp-spacy-parsed,GO-BP`

2.3 KB

By the KB, we mean a SPARQL endpoint populated with RDF statements which are results of conversion from the GE task results. To achieve the goal of establishing a seamless connection from the IE to KB, an automatic process is designed and implemented into PubAnnotation for:

- conversion of annotations to RDF statements, and
- feeding the statements into a SPARQL endpoint.

[2] Except for the coreference annotation, which is originally produced manually.

Also, a SPARQL-driven user interface to search the KB is designed and implemented.

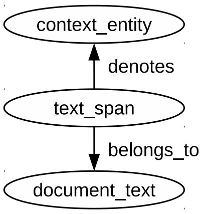


Figure 2: The core model of TAO

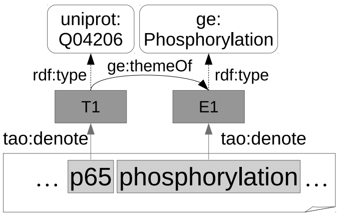


Figure 3: Annotation example using TAO

Considering its characteristics, the KB is designed to provide an easy access to the textual contexts of each knowledge piece. After surveying existing vocabularies for RDF statements (Ciccarese et al., 2011; Livingston et al., 2013), we chose to use a minimal vocabulary optimized for search, which we call *text annotation ontology (TAO)* (Kim et al., 2015). Figure 2 shows the core model of TAO, and Figure 3 shows an example of annotation representation using TAO. The example describes that

- the span *p65* "denotes" *T1*.
- *T1* is a *uniprot:Q04206*.
- the span *phosphorylation* "denotes" *E1*.
- *E1* is a *ge:Phosphorylation*.
- *T1* is a theme of *E1*.

Note that the role of TAO is to make connections between the two text spans, *p65* and *phosphorylation*, and the corresponding context entities, *T1* and *E1*, respectively[3]. Other parts of the annotations are described using other vocabularies: look at the two namespaces, *rdf* and *ge*.

A converter to produce RDF statements from annotations and a loader to feed the statements to a SPARQL endpoint is implemented to create an automatic flow from IE results to KB. TAO makes SPARQL queries to search the KB simple. For example, following query instructs the system to search for spans (*?s*) that denote an object (*?o*) which is a *uniprot:q04206*.

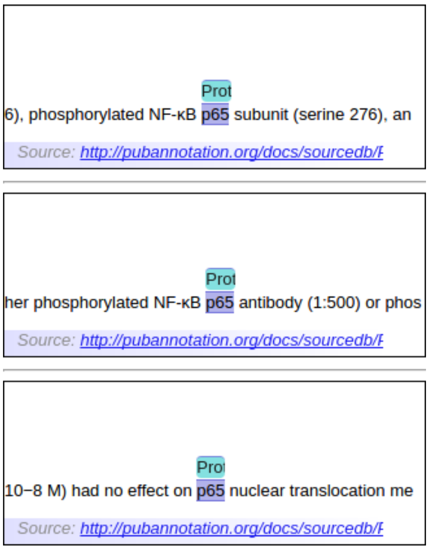
```
PREFIX tao:<http://pubannotation.org/ontology/tao.owl#>
PREFIX prj:<http://pubannotation.org/projects/>
PREFIX uniprot:<http://www.uniprot.org/uniprot/>

SELECT ?s
FROM prj:bionlp-st-ge-2016-uniprot
WHERE {
  ?s tao:denotes ?o .
  ?o a uniprot:Q04206 .
}
```

The results are URIs of the spans:

```
doc:sourcedb/PMC/sourceid/2664230/divs/2/spans/818-821
doc:sourcedb/PMC/sourceid/2664230/divs/5/spans/1128-1131
doc:sourcedb/PMC/sourceid/2674207/divs/18/spans/2512-2515
...
```

Note, however, that the span URIs are dereferenceable URIs which PubAnnotation provides. This means that the user can directly access the span following the URI. Figure 4 shows the spans of URIs from the above example rendered in TextAE[4], the default visualizer of PubAnnotation.



Prot
6), phosphorylated NF-κB p65 subunit (serine 276), an
Source: http://pubannotation.org/docs/sourcedb/F

Prot
her phosphorylated NF-κB p65 antibody (1:500) or phos
Source: http://pubannotation.org/docs/sourcedb/F

Prot
10−8 M) had no effect on p65 nuclear translocation me
Source: http://pubannotation.org/docs/sourcedb/F

Figure 4: Example of spans rendered in TextAE

2.4 Participation procedure

Participants to the GE4 task are supposed to go through following procedure:

1. To create a new project in PubAnnotation.
2. To import documents from the project, *bionlp-st-2016-test-proteins* to the new project. The 14 documents in the test set will be copied into the new project.
3. To import also annotations from the project, *bionlp-st-2016-test-proteins* to the project. All the protein annotations in the test set will be copied into the project.

[3] The prefixes, *T* and *E*, are used here just for readability. They do not hold any special meaning in the system.

[4] `http://textae.pubannotation.org`

4. At this point, the participant may want to compare the project against the test project. It will show that protein annotations are 100% correct, but the other annotations, e.g., events, are of 0%.
5. To produce event annotations, using a participating system, upon the protein annotations.
6. To upload the annotations to the project.
7. To compare the project against to the test project.

Every step of the procedure can be performed using the graphical interface of PubAnnotation. Some steps also can be performed using a programmable RESTful API of PubAnnotation. We believe the procedure is quite generic and can be applied to other shared tasks with similar setting.

3 Results and analyses

The results of GE4 organization are as follows:

- The general shared task framework implemented in PubAnnotation.
- The GE4 task re-engineered using the new framework
- The pipeline to populate a KB (SPARQL endpoint) from IE results
- The user interface to the KB
- The user experience of participants

As the first three are explained in previous sections, this section discusses the last two: KB user interface and user experience. Also, the benchmark data sets are analyzed to simulate the process of knowledge access using the KB.

3.1 User interface to IE-driven KB

A prototype interface to the IE-driven KB is implemented, of which a snapshot is shown in Figure 5. Since the KB is implemented as RDF data sets stored in a SPARQL endpoint, the interface is also SPARQL-oriented: see the input box for a SPARQL query in the center of the interface.

For those who are not familiar with SPARQL, a template system is implemented. A SPARQL template is a SPARQL query with placeholders, of which the value is easily changeable by user's input. For example, look at the template shown in Figure 6. It has one placeholder, __uniprot_id__. A placeholder is indicated by double underscore characters ('__') at its both sides. The title of the template is supposed to have the same placeholders. When displayed, the placeholders in the title become text input boxes to accept user's input, as shown at the top in the left pane of the screenshot. Upon change of the value in the input boxes, the placeholders in the SPARQL template are also updated, accordingly. Using the templates, users who are not familiar with SPARQL can still access the KB. Even for expert SPARQL users, it reduces time to author frequently necessary queries from scratch. In the left pane of the snapshot, 7 predefined templates are shown.

The next section presents results of analyzing benchmark data sets utilizing the templates.

3.2 Data analysis from KB perspective

In this section, the benchmark data sets are analyzed from a perspective of KB, and observations are discussed.

Table 1 shows statistics of UniProt ID annotations, which form the basis of the knowledge pieces of the KB we develop. For accuracy, only the UniProt ID annotations that are overlapping with (manual) protein annotations are counted. Note that, UniProt ID annotations that are not annotated as proteins in the benchmark data set are not involved in any further annotations, e.g. relations, so, anyway, they cannot be involved in any knowledge piece to be extracted from the data sets.

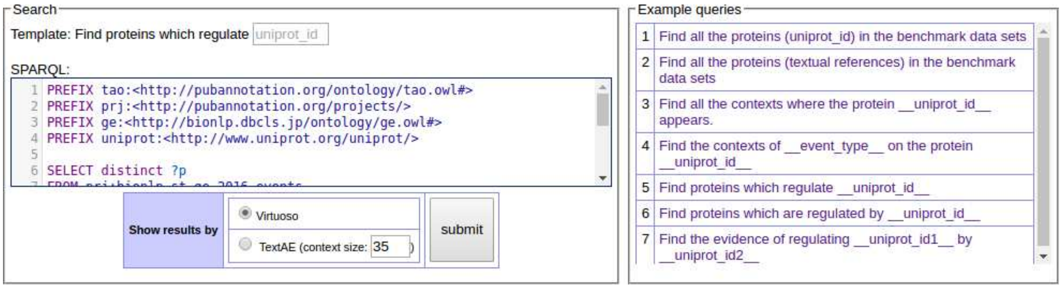
	Reference	Test	Sum
No. of instances	8,292	3,148	11,440
No. of types	221	110	242

Table 1: Statistics of UniProt ID annotation

Template 1, *Find all the proteins in the benchmark data sets*, with slight modifications, e.g. addition of *GROUP BY* modifier to count types, is used to obtain the statistics. Among the 110 UniProt IDs that appear in the test data set, 21 do not appear in the reference data set, simulating unseen protein names. They may represent an extra challenge for protein name recognition, and an extra chance for novel knowledge piece, at the same time.

Table 2 shows statistics of NFκB proteins, for which Template 3, *Find all the contexts where the protein __uniprot_id__ appears*, is used with the UniProt IDs of the 5 NFκB proteins set to the placeholder. It shows that *p65* is the most frequently referenced protein in both reference and test data sets.

One of typical search needs would be to find the proteins that regulate a certain protein, for which Template 5, *Find proteins which regulate*



Template: Find proteins which regulate uniprot_id

SPARQL:

```
PREFIX tao:<http://pubannotation.org/ontology/tao.owl#>
PREFIX prj:<http://pubannotation.org/projects/>
PREFIX ge:<http://bionlp.dbcls.jp/ontology/ge.owl#>
PREFIX uniprot:<http://www.uniprot.org/uniprot/>

SELECT distinct ?p
```

Show results by ● Virtuoso ○ TextAE (context size: 35)

1	Find all the proteins (uniprot_id) in the benchmark data sets
2	Find all the proteins (textual references) in the benchmark data sets
3	Find all the contexts where the protein __uniprot_id__ appears.
4	Find the contexts of __event_type__ on the protein __uniprot_id__
5	Find proteins which regulate __uniprot_id__
6	Find proteins which are regulated by __uniprot_id__
7	Find the evidence of regulating __uniprot_id1__ by __uniprot_id2__

Figure 5: SPARQL interface to the IE-induced KB

```
PREFIX tao:<http://pubannotation.org/ontology/tao.owl#>
PREFIX prj:<http://pubannotation.org/projects/>
PREFIX ge:<http://bionlp.dbcls.jp/ontology/ge.owl#>
PREFIX uniprot:<http://www.uniprot.org/uniprot/>

SELECT DISTINCT ?p
FROM prj:bionlp-st-ge-2016-events
FROM prj:bionlp-st-ge-2016-uniprot
WHERE {
  graph prj:bionlp-st-ge-2016-uniprot {
    ?o1 tao:denoted_by ?s1 .
    ?o1 a uniprot:__uniprot_id__ .
    ?o2 tao:denoted_by ?s2 .
    ?o2 a ?p .
  }

  ?o1_1 tao:denoted_by ?s1 .
  ?o2_1 tao:denoted_by ?s2 .
  ?o1_1 ^ge:partOf? / ge:themeOf+ ?e .
  ?o2_1 ^ge:partOf? / ge:causeOf+ ?e .

  FILTER (?p != tao:Context_entity)
}
```

Figure 6: A SPARQL template of title *Find proteins which regulate __uniprot_id__*

__uniprot_id__, can be used. With *Q04206* (p65) set to the placeholder, we find the following:

- In the ref. data, 21 proteins are found to regulate p65
- In the test data, 2 are found to regulate p65
- Among the 2 proteins found in the test data, one (P01375; TNFα) also in the reference data, whereas the other (P01584; IL1β) only in the test data.

Assuming that the reference data represents a KB at a point, and that the test data represents new feed to the KB, the piece of information that IL1B regulates p65 may represent a new piece of knowledge. On the other hand, the piece of information that TNFα regulates p65 itself may not represent a new knowledge. However, it may supply additional contexts to the known piece of knowledge, from which more detailed information, e.g. experimental condition, may be accessed.

Using Template 7, *Find the evidence for __uniprot_id1__ to regulate __uniprot_id2__*, with P01375 set to the first placeholder, and Q04206 to the second, we can access individual contexts of TNFα to regulate p65. Figure 7 shows one example, which suggests that more detailed knowledge about the regulation may be extracted by further digging the context, e.g., TNFα regulates phosphorylation of p65, and the specific sites of the phosphorylation are Ser529 and Ser536,

The series of analyses demonstrates that how IE results may contribute to populate the KB, and how the IE-driven KB can be explored using the template system.

3.3 Analyses on submissions

Due to the heavy burden of re-implementing the whole task, the GE4 task began as an experimental task. Many problems were encountered during the release of benchmark data sets and the evaluation system, which caused serious delay of the schedule. Thanks to voluntary comments and bug reports from some participants, most of the problems could be addressed, and, in the end, two systems were able to get through to the submission of results. However, as almost no time was given for the participants to adapt their systems to the task, submissions were made using the raw output from the systems, which caused the evaluation scores to be meaninglessly low. Thus, instead of reporting automatic evaluation results, we take the opportunity to discuss observations at the results.

Class	Uniprot ID	Name (Gene)	Reference	Test
I	P19838	Nuclear factor NF-kappa-B p105 subunit (NFKB1)	24	37
	Q00653	Nuclear factor NF-kappa-B p100 subunit (NFKB2)	8	12
II	Q04206	Transcription factor p65 (RELA)	295	98
	Q04864	Proto-oncogene c-Rel (REL)	16	6
	Q01201	Transcription factor RelB (RELB)	6	3

Table 2: Statistics of NFκb proteins in benchmark data sets

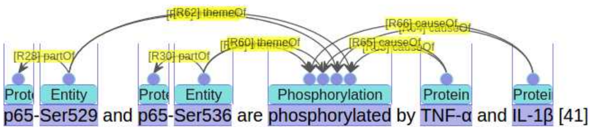


Figure 7: An annotation excerpt from PMC:3312845

One submission was made using the PKDE4J system (Song et al., 2015). An observation on the output revealed that a major discrepancy between the representation of GE4 and the system comes from the fact that while GE4 is an event extraction task PKDE4J is a relation extraction system. In other words, while GE4 requires events to be materialized in the representation, PKDE4J represents them as relations. An example shown in Figure 8 explains the difference. Note that the GE task materializes the events *Negative_regulation* and *Gene_expression* captured by the trigger words *inhibition* and *production*, respectively. While PKDE4J does not materialize them, however, it correctly extracts the relation that *IL-10* down-regulates *interferon gamma*. It also correctly extracts the relation that *IL-10* down-regulates *suppressor of cytokine signaling I*. Although PKDE4J does not recognize the *Negative_regulation* captured by *Resistance*, it seems right considering that PKDE4J is a relation extraction system which requires two arguments for each relation. The observation suggests that characteristics of individual systems need to be carefully considered to better understand and utilize them.

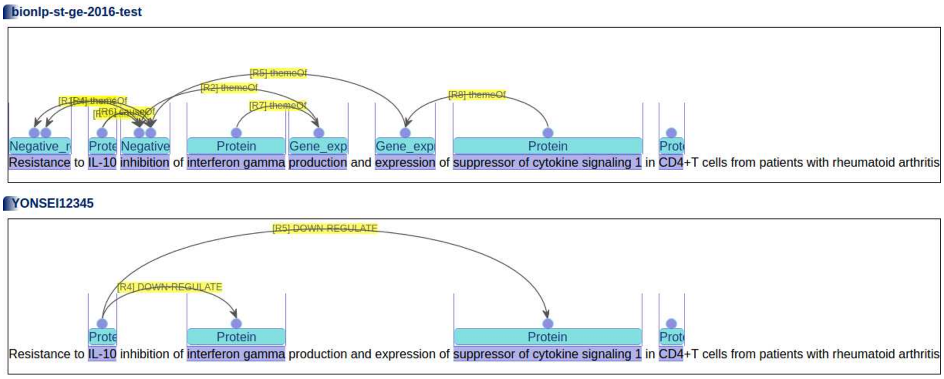
Furthermore, an attempt was made to use TEES, an open source event extraction system, which won previous iterations of the GE task (Björne and Salakoski, 2013). The goal was to observe TEES' out-of-the-box performance in the GE4 task. With TEES, a different way of entering the task, namely submission of the URL of a RESTful web service, was tested. PubAnnotation offers a function to communicate with a web service to obtain annotations from it. Thus, by submitting the URL of an annotation system which implements a RESTful API, annotations can be pulled into PubAnnotation. In order to make use of this feature, a small script was written that runs TEES as a RESTful web service, and annotations obtained directly through PubAnnotation. Conversion from the Interaction XML, TEES' native format, to the PubAnnotation JSON format was only minimally implemented, to test the submission. To make use of the performance of TEES, the conversion needs to be implemented more thoroughly, which is left as a future work.

4 Conclusions

The GE4 task is organized as an experimental task, toward generalization of the shared task resources and seamless connection of IE task results to KB population. As the result, a new shared task system is implemented using PubAnnotation as the platform. Note that PubAnnotation itself is an open source project. By being embraced by the open platform, the shared task system is expected to become more sustainable, and accessible. As the newly implemented system is fairly generic, organizing a new shared task is easy, which we hope to promote organization of more shared tasks by interested parties, particularly by domain experts. The GE4 shared task will be running continuously inviting open participation from the community.

Acknowledgments

The work is supported by Life Science Database Integration Project funded by National Bioscience Database Center (NBDC) of Japan Science and Technology Agency (JST).


bionlp-st-ge-2016-test

Resistance to IL-10 inhibition of interferon gamma production and expression of suppressor of cytokine signaling 1 in CD4+T cells from patients with rheumatoid arthritis

YONSEI12345

Resistance to IL-10 inhibition of interferon gamma production and expression of suppressor of cytokine signaling 1 in CD4+T cells from patients with rheumatoid arthritis

Figure 8: Example of the output of PKDE4J system

References

Jari Björne and Tapio Salakoski. 2013. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25, Sofia, Bulgaria, August.

J Gregory Caporaso, J Lynn Fink, Philip E Bourne, K Bretonnel Cohen, and Lawrence Hunter. 2008. Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. *Pacific Symposium on Biocomputing*, pages 640–6513.

Hong-Woo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii. 2006. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, pages 4–15, Maui, Hawaii, USA, January.

Paolo Ciccarese, Marco Ocana, Leyla Garcia Castro, Sudeshna Das, and Tim Clark. 2011. An open annotation ontology for science on web 3.0. *Journal of Biomedical Semantics*, 2(Suppl 2):S4.

Matthew Honnibal, Yoav Goldberg, and Mark Johnson. 2013. A non-monotonic arc-eager transition system for dependency parsing. In *CoNLL*, pages 163–172.

Jin-Dong Kim and Yue Wang. 2012. Pubannotation: A persistent and sharable corpus and annotation repository. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, pages 202–205, Stroudsburg, PA, USA.

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) Workshop*, pages 1–9.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. 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.

Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. The Genia Event Extraction Shared Task, 2013 Edition - Overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15, Sofia, Bulgaria, August.

Jin-Dong Kim, Jung-jae Kim, Xu Han, and Dietrich Rebholz-Schuhmann. 2015. Extending the evaluation of genia event task toward knowledge base construction and comparison to gene regulation ontology task. *BMC Bioinformatics*, 16(Suppl 10):S3.

Martin Krallinger, Florian Leitner, and Alfonso Valencia. 2007. Assessment of the Second BioCreative PPI task: Automatic Extraction of Protein-Protein Interactions. In L. Hirschman, M. Krallinger, and A. Valencia, editors, *Proceedings of Second BioCreative Challenge Evaluation Workshop*, pages 29–39.

Kevin Livingston, Michael Bada, Lawrence Hunter, and Karin Verspoor. 2013. Representing annotation compositionality and provenance for the semantic web. *Journal of Biomedical Semantics*, 4(1):38.

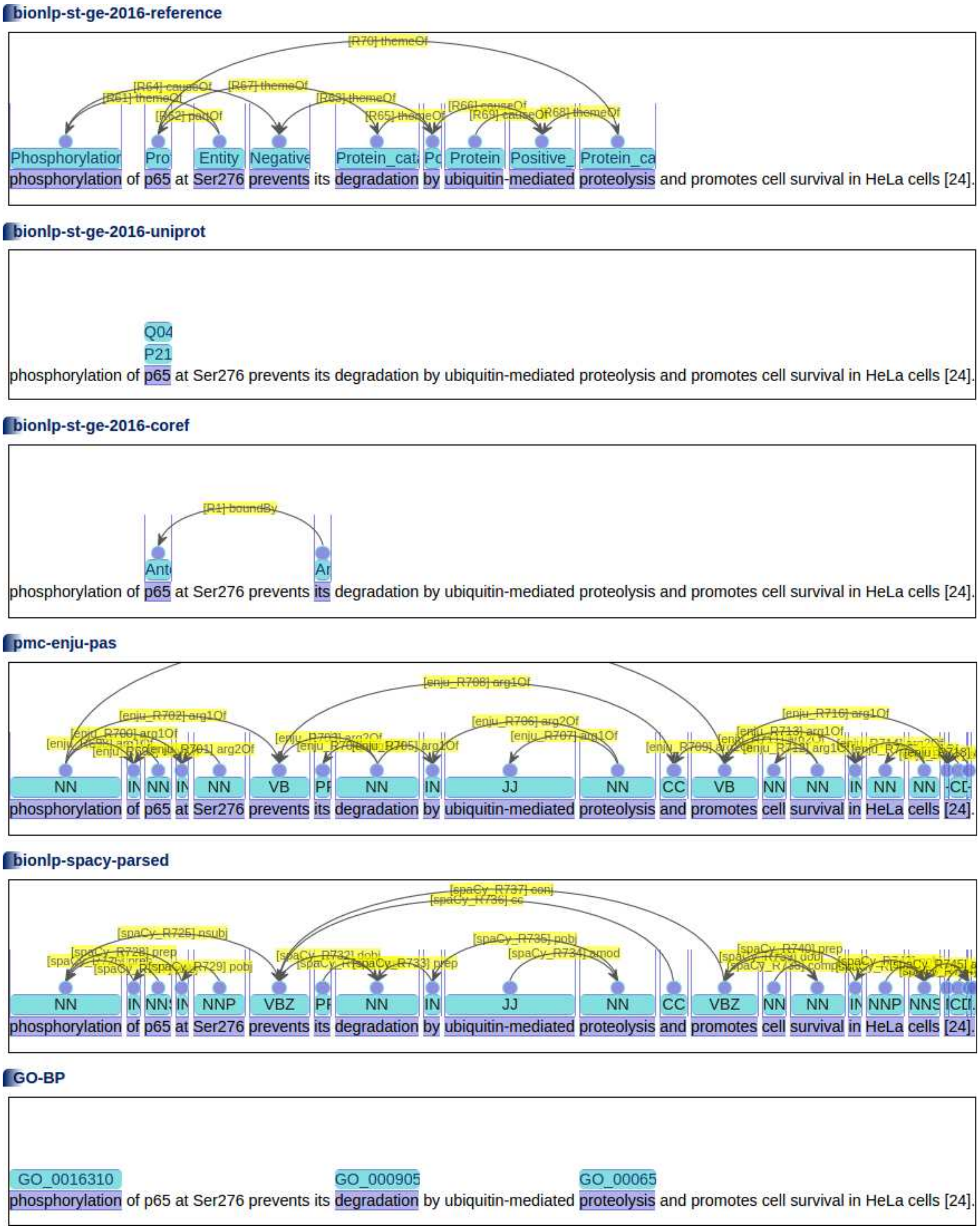
Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner. 2004. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4):547–556.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun'ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology (JBCB)*, 8(1):131–146, February.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic hpsg parsing. *Comput. Linguist.*, 34(1):35–80, March.

Christopher J Mungall, Carlo Torniai, Georgios V Gkoutos, Suzanna E Lewis, and Melissa A Haendel. 2012. Uberon, an integrative multi-species anatomy ontology. *Genome Biology*, 13(1):R5.

Min Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang. 2015. Pkde4j: Entity and relation extraction for public knowledge discovery. *Journal of biomedical informatics*, 57:320–332.



phosphorylation of p65 at Ser276 prevents its degradation by ubiquitin-mediated proteolysis and promotes cell survival in HeLa cells [24].

Figure 9: Excerpts of annotation data sets for the GE4 task