Typed Entity and Relation Annotation on Computer Science Papers

Yuka Tateisi[1], Tomoko Ohta[2], Yusuke Miyao[3], Sampo Pyysalo[4], Akiko Aizawa[3]
[1]Japan Science and Technology Agency, [2]textimi, [3]National Institute of Informatics, [4]University of Cambridge
E-mail: tateisi@biosciencedbc.jp, tomoko.ohta@textimi.com, {yusuke,aizawa}@nii.ac.jp, sampo@pyysalo.net

Abstract

We describe our ongoing effort to establish an annotation scheme for describing the semantic structures of research articles in the computer science domain, with the intended use of developing search systems that can refine their results by the roles of the entities denoted by the query keys. In our scheme, mentions of entities are annotated with ontology-based types, and the roles of the entities are annotated as relations with other entities described in the text. So far, we have annotated 400 abstracts from the ACL anthology and the ACM digital library. In this paper, the scheme and the annotated dataset are described, along with the problems found in the course of annotation. We also show the results of automatic annotation and evaluate the corpus in a practical setting in application to topic extraction.

Keywords: corpus creation; semantic relation; information extraction from scientific text

1. Introduction

Surveying technical documents such as research papers and patents is a critical task for research scientists, administrators, and policy makers. The automated analysis of technical documents is expected to assist them by retrieving relevant documents and extracting information of interest. In this context, intelligent content-based search systems that can answer queries such as *What tasks have CRFs been used for?* and *What methods have been used for POS tagging?* are in demand. For these queries, traditional keyword-based systems are insufficient, as they can only search for the mention of an entity represented by the keywords. These queries search for entities playing a particular role in some context, e.g., CRF as a method in the query *What tasks have CRFs been used for?*

Answers to these queries can be found in various forms in published papers, for example, as phrase-internal structures (*CRF-based POS tagging*), as sentence-internal structures (*CRFs have been successfully applied to POS tagging*), and as inter-sentential discourse structures (*In this study we propose a new method for the efficiently training CRFs. The proposed method is evaluated for POS tagging tasks*). Thus, as several layers of linguistic structure must be investigated in order to answer such queries, we aim to establish a framework for uniformly representing semantic structure involving the roles of entities described in technical documents, represented across those layers.

The roles of entities are determined in the context of an event that they are involved in, so that an event-based annotation framework such as the one used in annotating articles in the biomedical domain (e.g., Kim et al., 2008) would be suitable for the current purpose. However, in-domain event framesets in the computer science/technology domain are yet to be established. In addition, because computers and computational methods can be applied to a wide and ever-widening range of topics, formalizing the frameset for events in the computer science/technology domain is an extremely difficult task.

Therefore, instead of precisely defining a frameset for in-domain events, we describe the roles of entities in the form of their mutual relations using a set of general relationships such as method-purpose, system-output, and evaluation-result.

In addition, we adopt a classification scheme of entities based on their intrinsic nature by anchoring the entity mentions to ontology-based types. This is a common approach taken in biomedical corpus annotation but has not been done for computer science/technology corpus. In this paper, we describe the annotation scheme and initial annotation results, and investigate the effect of entity typing along with the problems in annotation resulting from it. We also evaluate the corpus in a practical setting in application to topic extraction.

2. Related Work

The focus of research on searching research papers has been shifting from the social aspects of papers and their authors, such as citation link analysis and co-authorship analysis implemented in search engines such as Google Scholar [1], to more content-based analysis such as information extraction (IE) concerning the methodological aspects of research papers and patents for analyzing technical trends and discovering emerging research fields. Their focus is on determining how things such as systems and data are developed and used. Consequently, in the annotated corpora used for establishing the systems for these purposes, things described in a document are labeled and classified according to their role in a certain context, such as application domain, method, and product.

Some studies attach role-based labels to entity mentions. For example, Gupta and Manning (2011), in establishing a method for identifying the technical trends from abstracts in the ACL anthology [2], extracted the FOCUS (main contribution of the article), DOMAIN (application domain), and TECHNIQUE (a method or tool used to achieve the FOCUS). The corpus used for the study attaches these labels directly to mentions of the corresponding entities. Similarly, Fukuda et al. (2012) annotated and classified entities in patent documents as TECHNOLOGY

[1] https://scholar.google.com

[2] https://aclweb.org/anthology/

Type	Definition	Example
THING	`Thing` (The top level)	
OCCURRENT	`Occurrent`	
PROCESS	`Processual Entity`	*running,computation*
TIME	`Temporal Region`	*2012, before, waiting time*
CONTINUANT	`Continuant`	
ARTIFACT	Subclass of `Object`: physical object created for a purpose	*mobile devices, Mac*
DATA-ITEM	`Data Item` and `Textual Entity` in IAO	*lower bound, cost, sentence*
LOCATION	`Spatial Region`	*Asia, space, between*
PERSON	Subclass of `Object`: individual or group of people	*human, Eugene Charniak*
PLAN	`Processual Entity`	*CRF, algorithm*
QUALITY	`Quality`	*qualitative, new*
QUANTITY	Numbers, with or without units	*five, two-fold, several*
MODALITY	Modality	*can, cannot, need to*
REFERENCE	Anaphoric expressions	it, they
EXTERNAL-REFERENCE	Literature reference (citation)	*Miyao and Tsujii 2008, [1]*
LANGUAGE	Languages for inter-human communication	*English, natural language*
DOMAIN	Areas of study	*NLP, biomedicine*
ORGANIZATION	Group of people established for a purpose	*ERLA, universities*
FORMULA	Mathematical formula	*F=0.98*
PLAN-OR-PROCESS	See the main text	
JUDGING-PROCESS		
INTELLIGENT-AGENT		

Table 1: Entity tags, definitions and examples: names in monospaced font denotes the class in IAO

(algorithms, materials, tools, and data used in invention), EFFECT (effects of a technology that can be expressed as a pair comprising an attribute and a value), and ATTRIBUTE and VALUE (attribute and value in the effect). Anick et al. (2014) extracted technology terms defined as Artifact (object created as a result of some process), Process/Technique (method for creation) or Field (a discipline or a scientific area relating to creation) using a corpus in which mentions of entities playing these roles are labeled. Roth and Klein (2015) extracted terms that denote an ACTION, ACTOR, OBJECT, and PROPERTY, using an annotated dataset in which entity mentions are labeled based on the ontology defined by Roth et al. (2014). In their ontology concepts are classified according to roles that things can play in a particular operation, such as a participant, actor, object, and property.

Another type of approach to capturing the structure of entity roles is to annotate the relationship between entities to label the entities as "things in a certain context" and "how they are related to other things in the same context". Kameda et al. (2013), using Related Work sections from the proceedings of the Association for the Advancement of Artificial Intelligence (AAAI2010), identified the paper-topic relation along with the method-purpose relation among concepts described in the paper in order to construct a network representing the methods developed in one study and used by others and to evaluate the influence of the research. Nassour-Kassis et al. (2015) identified the mentions of tasks and attributes and linked them with one of 6 types (Means-End, Instance-of, Consists-of, Associated-with, Contributes-to, and Compares-to) of relations, using ten articles on summarization for building a conceptual map in the natural language processing domain. Tateisi et al. (2014) developed a corpus on research articles from Journal of Information Processing Society of Japan (IPSJ Journal) where relationship among OBJECTS (named entities), MEASURE (judgment and evaluation, including numbers), and TERM (general technical concepts other than OBJECT and MEASURE) are identified and labeled with one of 16 types such as Apply-to (method-purpose), Evaluate (evaluation object-evaluation result), and Attribute (object-attribute), and developed a prototype of a keyword-based search system in which results can be filtered according to the relations involving the keyword. Those works do not investigate the types of the entities themselves and have very a shallow classification of entity types.

In the current work, we basically follow the latter approach, but incorporate entity typing based on the nature of the entities. Entity typing enables annotators to help find the type restrictions in relations arguments and validate relation annotations. We also believe that the classification of entities will help to establish in-domain lexicons and event frames and enable typed inferences. As far as we know, there is no previous work that incorporates nature-based, as opposed to role-based, entity typing in the annotation of documents in the computer science/technology domain.

3. Annotation Scheme

Our aim is to develop a corpus to identify technical entities, their natures, and the roles they are playing in the context of the work described in a research article in computer science/technology. We intend the corpus to be used for more general information extraction tasks than simply extracting fixed kinds of relations such as method-purpose.

Type	Definition	Example
APPLY-TO(*A*, *B*)	A method *A* is applied to achieve the purpose *B*	*CRF* $_{A}$-based *tagger* $_{B}$
RESULT(*A*,*B*)	*B* is a logical conclusion or an unintended result of *A*	*Multi-modal interface* $_{A}$ led to *3.5fold speed improvement* $_{B}$
AGENT(*A*,*B*)	*B* is the intentional (or seemingly intentional) agent of a process *A*	a frustrated *player* $_{B}$ of a *game* $_{A}$
INPUT(*A*,*B*)	*B* is the input of a system or a process *A*; *B* is consumed by *A*	*corpus* $_{B}$ for *training* $_{A}$
OUTPUT(*A*, *B*)	*B* is the output of a system or a process *A*; *B* is generated by A	an *image* $_{B}$ is *displayed* $_{A}$
IN_OUT(*A*, *B*)	*B* is simultaneously INPUT and OUTPUT and is changed by a system or a process *A*	a *modified* $_{A}$ *annotation schema* $_{B}$
TARGET(*A*, *B*)	*B* is the target of an action *A*, which does not change	to *drive* $_{A}$ a *bus* $_{B}$
ORIGIN(*A*, *B*)	*B* is the starting point of action *A*	the *projcect* $_{B}$ started in *2011* $_{A}$
DESTINATION(*A*, *B*)	*B* is the ending point of action *A*	an image *displayed* $_{A}$ on a *palm* $_{B}$
ORI_DEST(*A*, *B*)	*B* is the starting and ending point of a single action *A*	*oscillate* $_{A}$ between *two numbers* $_{B}$
CONDITION(*A*, *B*)	The condition *B* holds in situation *A*	a *survey* $_{A}$ conducted in *India* $_{B}$
ATTRIBUTE(*A*, *B*)	*B* is an attribute or a characteristic of *A*	*accuracy* $_{B}$ of the *tagger* $_{A}$
POSS(*A*, *B*)	*A* is owned by *B*	*LDC* $_{B}$'s *corpora* $_{A}$
COMPARE(*A*, *B*)	*A* is compared to *B* in evaluation	*F–score* $_{A}$ compared to the *baseline* $_{B}$
IS-A(*A*,*B*)	*A* is a hypernym of *B*	*services* $_{A}$ *such as Google* $_{B}$
MEMBER-COLLECTION(*A*, *B*)	*B* is a member of *A*	a *sentence* $_{B}$ in *PTB* $_{A}$
COMPONENT-OBJECT(*A*, *B*)	*B* is a component of *A*	a *back button* $_{B}$ in the *toolbar* $_{A}$
EQUIVALENCE(*A*, *B*)	Locally-defined synonymy between *A* and *B*	*DoS* $_{B}$ (*denial–of –service* $_{A}$) attack
COREFERENCE(*A*, *B*)	Anaphora *A* and antecedent *B*	retrieve the *documents* $_{B}$ and store *them* $_{A}$
SPLIT(*A*,*B*)	Denotes a multiword expression split by parenthetic expression	*DoS* $_{B}$ (denial–of –service) *attack* $_{A}$

Table 2: Relation tags, definitions and examples

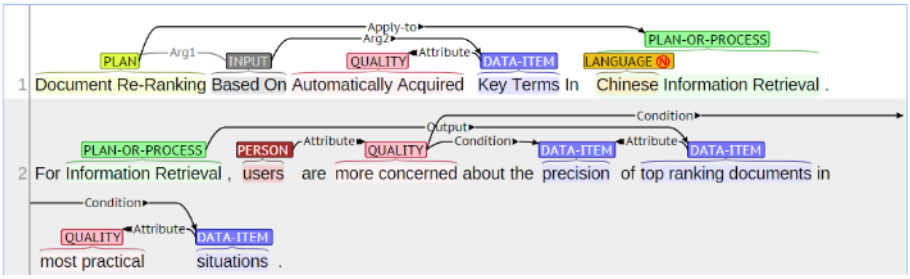

Figure 1: Annotation example (ACL anthology C04-1069) shown in brat rapid annotation tool

Thus, we attempt to capture the relations involving the entities within sentences in the articles as completely as possible.

For example, in sentences such as *CRF-based POS tagging has achieved state-of-the-art accuracy*, we can recognize various fragments of information in addition to "CRF is used for POS tagging". Specifically, we can read that the state-of-the-art accuracy is achieved for POS tagging. As such information can be an answer to different search requests, and there could be yet other information to be searched, we believe annotating the relationship described in the sentences as completely as possible would be useful for a variety of purposes. As a result, we have decided to annotate relations other than the ones for roles, including discourse-oriented relations such as cause-result and ontological relations such as hypernym-hyponym.

Our annotation scheme is an extension of that proposed by (Tateisi et al., 2014), to which we add a classification scheme for entities. For this purpose, we use the Information Artifact Ontology (IAO) (Ruttenberg, 2014), which is an ontology for technological objects based on the top-level Basic Formal Ontology, and used in bioinformatics area to describe the experimental procedures, data sets, and technical instructions.

In our annotation, text spans that mention named and other technical entities, including operations and events, are identified. The spans are labelled with one of the types derived from IAO or types added as necessary. Then, the relationships among them are identified and annotated in the form of directed, typed binary relations.

We incorporated the following types from IAO: `occurrent` (an entity that has temporal parts); `processual entity` (an entity that can exist in time by occurring or happening); `temporal region` (a part of time); `continuant` (an entity that exists in full at any time and has no temporal parts); `spatial region` (a region in space that inherits no other entities); `object` (an independent physical entity, including an animal); `directive information entity` (an information content entity that, under certain interpretation, is a directives for undertaking a process); `data item` (an information content entity generically dependent on some

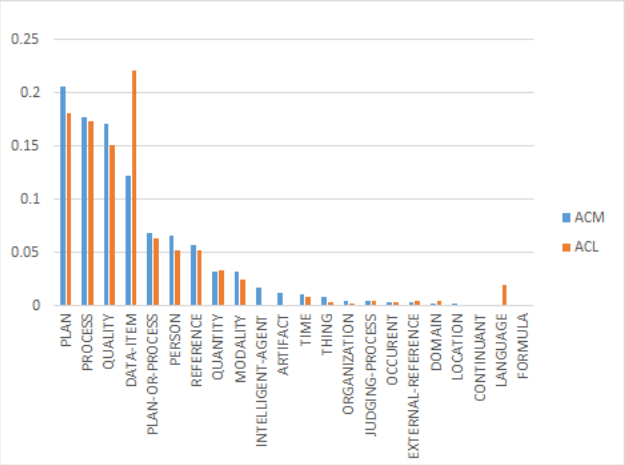

Figure 2: Distribution of entity types

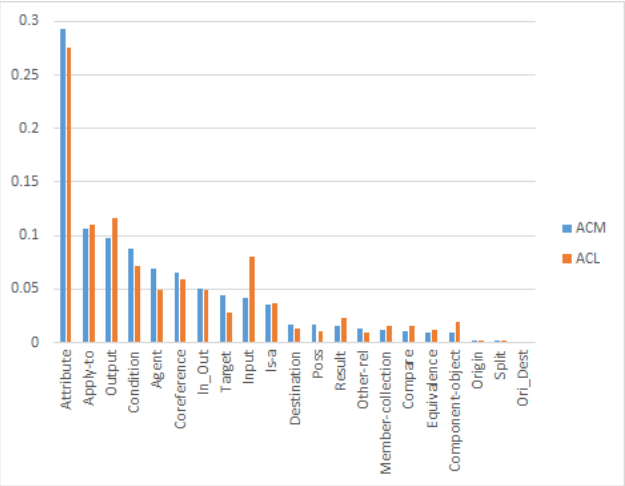

Figure 3: Distribution of relation types

artifact and intended to be a truthful statement regarding the artifact); `textual entity` (a pattern of glyphs intended to be interpreted); and `quality` (a property of other entities).

For annotation simplicity, we (1) renamed `processual entity`, `temporal region`, and `spatial region` to PLAN, TIME, and LOCATION, respectively, (2) divided `object` into ARTIFACT (a physical object intentionally created for a purpose) and PERSON (an individual or a group of people), and (3) merged `data item` and `textual entity` (i.e., non-directive information entities) to DATA-ITEM.

We also added the following types not defined in IAO to capture the phenomena described in scientific texts, on a data-driven basis: QUANTITY (a number with or without units); MODALITY (modality); REFERENCE (an anaphoric expression); EXTERNAL-REFERENCE (a literature reference); LANGUAGE (a language for human-human communication); ORGANIZATION (a group of people established for a purpose); DOMAIN (an area of study); and FORMULA (a mathematical formula).

In addition, we defined the following compound or "ambiguous" types to handle systematic ambiguity in natural language text where the distinction is unclear and unnecessary for practical purposes: PLAN-OR-PROCESS (an expression such as "*web search*" that can denote a process, a function that realizes the process, or steps of instructions to achieve the function), INTELLIGENT-AGENT (an expression that can be interpreted as people or artifacts/programs that emulate human behavior, e.g., *players* (of video games)), and JUDGING-PROCESS (an expression that describes a system's behavior and also the author's subjective judgment, e.g., *outperform* in "*The current system outperforms the baseline*"). Table 1 summarizes the tag set for entity annotation.

The relations between entities are also typed based on the classification used for annotating IPSJ abstracts (Tateisi et al., 2014), given in Table 2 with examples. In addition to those shown in Table 2, we used a catch-all relation OTHER-REL to annotate relations that annotators recognized but could not assign to any of the pre-defined relation types. For simplicity, we only annotate intra-sentence relations with exception of COREFERENCE, which may be annotated across sentences. Different from the original scheme, we also decided to annotate words and phrases that express the relations (relation triggers) as much as possible, except in cases where triggers are prepositions or punctuation marks.

4. Annotated Data

Our dataset was constructed from 400 abstracts of research papers (250 abstracts from the ACL anthology and 150 from the ACM digital library[3]). In the ACL subset, 150 abstracts were randomly selected from the entire set and the remaining 100 were randomly selected from the set used by Gupta and Manning (2011). The abstracts in the ACM subset were randomly selected from the set used for the SEMEVAL-2010 task 5 (Kim et al., 2010). Errors in text resulting from PDF conversion were manually corrected. Annotation was performed by a single annotator (the second author). A screenshot of the brat system (Stenetorp et al., 2012) is given in Figure 1 as an annotation example. In 1959 sentences in the ACL set, 14887 entities and 13310 relations were identified. In the 1213 sentences in the ACM set, the numbers of identified entities and relations were 12463 and 11201, respectively. The distributions of entity and relation types, in proportion, in the two domains are shown in Figures 2 and 3.

The results shown in Figure 2 indicate that software (PLAN, PLAN-OR-PROCESS) is more frequently discussed than hardware (ARTIFACT) in both the general computer science/technology domain (ACM) and the natural language processing subdomain (ACL), but the tendency is more prominent in ACL. ACL is also characterized by a larger proportion of data (DATA-ITEM) being discussed than in ACM, and, as expected, natural languages (LANGUAGE) appear more frequently in ACL. On the other hand, the items labeled INTELLIGENT-AGENT characterize the ACM set, owing to several articles about electronic commerce. These observations indicate that even

[3] http://dl.acm.org/

Class	#Occ	P	R	F	Confused Classes	#Miss
DATA-ITEM	1120	0.70	0.59	0.64	PLAN(0.07), QUAL(0.04), POP(0.02)	262
PLAN	929	0.64	0.66	0.65	DATA(0.05), POP(0.05), PROC(0.02), ART(0.01)	174
PROCESS	892	0.85	0.81	0.83	POP(0.01), PROC(0.01), J-P(0.01)	124
QUALITY	682	0.69	0.69	0.69	DATA(0.02), J-P(0.02), PROC(0.02)	161
PLAN-OR-PROCESS	336	0.51	0.45	0.48	PLAN(0.18), DATA(0.10), PROC(0.04)	76
PERSON	257	0.92	0.84	0.88	DATA(0.01), ORG(0.01), PLAN(0.01)	24
REFERENCE	253	0.91	0.91	0.91		18
QUANTITY	148	0.78	0.82	0.81		24
LANGUAGE	121	0.71	0.39	0.5	DATA (0.01)	68
MODALITY	101	0.84	0.86	0.85	DATA(0.01), PROC(0.01), QUAL(0.01)	11
TIME	38	0.56	0.34	0.41	MOD(0.05),	16
DOMAIN	26	0.48	0.46	0.47	PLAN(0.23), PROC (0.04)	7
JUDGING-PROCESS	21	0.35	0.76	0.48	PROC (0.05), QUAL (0.05)	3
EXTERNAL-REFERENCE	16	0.22	0.44	0.29		9
ORGANIZATION	7	0	0	0	DATA(0.43), I-A(0.14), PLAN(0.14), POP(0.14)	1
LOCATION	1	0.5	1	0.67		0
ARTIFACT, FORMULA, INTELLIGENT-AGENT, OCCURRENT, THING	0					

Table 3: Results for entity extraction. (ART:ARTIFACT, DATA:DATA-ITEM, I-A:INTELLIGENT-AGENT, J-P:JUDGING-PROCESS, MOD:MODALITY, ORG:ORGANIZATION, POP:PLAN-OR-PROCESS, PROC:PROCESS, QUAL:QUALITTY)

shallow classification with a top-level ontology can capture the characteristics of research subdomains.

The distribution of relation types, shown in Figure 3, are more similar in the two subdomains than the distribution of entity types, except that the INPUT and OUTPUT relations are more frequent in the ACL subdomain. This also indicates that the ACL subdomain is more data-oriented.

A notable characteristic of the distribution of relation types is that the ATTRIBUTE relation is very frequent, almost as three times as frequent as the second-most frequent types (APPLY-TO in ACM and OUTPUT in ACL). This shows that properties of things are frequently described in research papers. It also indicates that the granularity of the scope of ATTRIBUTE relation is wider than that of others, i.e., properties can further be broken into several subtypes.

5. Entity Typing and Type Restriction of Relation Arguments

We derived several heuristic rules for restricting the argument types of relations, e.g., "the two arguments of the IS-A relation must be of the same type" (Rule-IS), "the arguments of the APPLY-TO (method and purpose) relation must be of PROCESS, PLAN, PLAN-OR-PROCESS, or REFERENCE type" (Rule-APP), and "the second argument of the INPUT, OUTPUT and IN_OUT relation must be of DATA-ITEM, QUANTITY, QUALITY, PROCESS, PLAN, PLAN-OR-PROCESS, or REFERENCE type" (Rule-IO), and found 127 violations. Through examining the violations, we have found issues in the annotation scheme and the ambiguity/metonymy treatment.

A prominent problem was related to the words denoting abstract roles such as *feature* and *component* and the entity playing the role in a certain context. Consider the sentence *The model consists of three main components: (i) a lexicon, (...) (ii) a rewrite rules component (...) and (iii) a morphotactic component (...).* The three components mentioned are entities of different types, i.e., *lexicon* is a dataset (DATA-ITEM), and the others are program functions (PLAN). The current convention uses IS-A relation to relate *components* and *lexicon* etc., which is impossible without violating Rule-IS. This suggests that we need to define a new relation for role-playing and a new type or types for "role" words such as *components.*

We also found that ambiguity and metonymic constructions cause annotation difficulty. These violations suggest a need for a type-coercion mechanism, such as dot-types (Pustejovsky et al. 2009).

For example, when a process uses parameters, the names of the parameters can denote "the invocation of the process with the parameters" (e. g. *RM pairs extracted can perform the mapping*, where *RM pairs extracted* denotes a process using the pairs as parameters) and the name of data structure is used for both the data structure itself and the content of the data (*Bigrams and trigrams are commonly used in statistical natural language processing*). They lead to the annotation of APPLY-TO relation between DATA-ITEM and PROCESS, which violates Rule-APP.

Another type of the problem is the ambiguity between an entity of any type and the data about its features, especially in statements concerning information extraction (IE), leading to violation of Rule-IO. For example, in *we can retrieve Eugene Charniak via search for statistical parsing,* the name *Eugine Charniak* does not denote Dr. Charniak himself but bibliographic data concerning his work. Even outside of an IE context, it is not unusual to denote the numeric data concerning an entity using the name of the entity itself, e.g., *reduces the operations of the generator*

Types	#Occ	P	R	F	Confused Types	#Miss
Attribute	1122	0.64	0.57	0.6	Apply-to(0.01), Poss(0.01)	380
Output	564	0.58	0.45	0.51	Input(0.04), In_Out(0.02)	229
Apply-to	488	0.43	0.41	0.42		248
Condition	379	0.47	0.27	0.34	Attribute(0.05), Apply-to(0.01),	207
Input	348	0.38	0.31	0.34	In_Out(0.31), Output(0.02), Apply-to(0.01)	195
Agent	258	0.86	0.73	0.79	Output(0.01)	48
In_Out	249	0.49	0.39	0.43	Output(0.12), Input(0.06), Condition(0.02), Apply-to(0.01), Target(0.01)	80
Coreference	247	0.5	0.32	0.39		164
Is-a	206	0.32	0.11	0.16		157
Target	155	0.47	0.24	0.32	Output(0.08), In_Out(0.06), Input(0.05), Apply-to(0.05), Attribute(0.03), Destination(0.01)	68
Result	114	0.35	0.16	0.22	Apply-to(0.09), Output(0.02)	79
Component-object	102	0	0	0	Output(0.03), Mem-Col(0.02), Apply-to(0.02), Attribute(0.02)	65
Destination	72	0.26	0.14	0.18	Condition(0.14), Input(0.11),Apply-to(0.03), In_Out(0.01)	34
Equivalence	60	0.35	0.33	0.34	Comp-Obj(0.02)	37
Compare	53	0.38	0.42	0.4		24
Member-collection	46	0.2	0.37	0.26	Attribute(0.06)	18
Poss	44	0.51	0.8	0.62		7
Origin	6	0	0	0	Input(0.33), Condition(0.17), Output(0.17)	2
Ori_Dest	2	0	0	0	Comp-Obj (0.5)	1
Other-rel	9	0	0	0		9

Table 4: Relation extraction results (Comp-Obj:Component-Object, Mem-Col:Member-Collection)

where *operations* denotes the number of the operation instances, and *obtaining the locations of sensor nodes* where l*ocations* denotes the location coordinates.

6. Automatic Entity and Relation Extraction

We trained the entity-relation extraction model of Miwa and Sasaki (2014) to annotate texts automatically using our scheme. The method is a history-based structured learning approach that jointly extracts entities and relations and maps the extraction task to a filling problem of a table that represents them jointly. The method has been reported to have achieved precision, recall, and F1 score of 0.837, 0.599, and 0.698, respectively, for relation extraction from the CONLL-2004 dataset (Roth and Yih 2004), significantly outperforming conventional pipeline approaches.

For training the model, we used the same features used in the original model, from syntactic parsers Enju (Miyao and Tsujii, 2008) and LRDEP (Sagae and Tsujii, 2007). We utilized perceptron with a max-violation update for the learning method and close-first/right-to-left for the table search order. See Miwa and Sasaki (2014) for a detailed description of the parameters.

Two hundred and fifty (250) abstracts were randomly selected from the corpus excluding those taken from the Gupta-Manning set. Using 10-fold cross validation, (precision, recall, F1) for entities and relations are (0.629, 0.628, 0.629) and (0.543, 0.452, 0.493), respectively. A relation is judged correct when the type, direction, and the last tokens of the related entities are correct.

The acquired model was also applied to the 100 abstracts in our corpus from the Gupta-Manning set. Overall, the results were slightly better than the cross-validation results, with (precision, recall, F1) being (0.680, 0.706, 0.693) for entities and (0.416, 0.523, 0.463) for relations. Tables 3 and 4 show the results for each of the entity classes and each of the relation classes. In these tables, numbers in the #Occ and the #Miss columns are, respectively, the numbers of entities/relations manually annotated in the 100 abstracts and those of entities/relations that the automatic annotation failed to find at all, P, R, and F are precisions, recall, and F-1 score, respectively, and Confused types are frequent (>= 1%) types erroneously assigned by automatic annotation. As seen in the tables, precision is relatively greater for most of the entities/relations.

The results in Table 3 indicates the tendency for the automatic annotation to assign DATA-ITEM, PLAN, and PROCESS labels, which are more frequent than others in ACL according to the distribution shown in Figure 2 in Section 4. The results also support two of our observations on type ambiguity. One is concerning the ambiguities we expected in designing the scheme and resulted in our decision to incorporate the "ambiguous/compound" types: PLAN/PROCESS/PLAN-OR-PROCESS and PROCESS/QUALITY/JUDGING-PROCESS. The confusion indicates that they are indeed difficult to distinguish. The other ambiguity suggested by the results is the confusion involving DATA-ITEM and PLAN, suggesting the abundance of issues similar to those discussed in the previous section.

The results in Table 4 show similar confusion patterns to those observed in inter-annotator results in the Japanese version (Tateisi et al. 2013) such as APPLY-TO/INPUT/OUTPUT and ATTRIBUTE/CONDITION. This suggests a language-independent difficulty in

annotating using this scheme.

7. Application to Topic Extraction

We evaluated the annotation in a practical settings to confirm that the relation structure is useful for applications. For this purpose, we attempted to extract FOCUS, DOMAIN, and TECHNIQUE of Gupta and Manning (2011). This experiment corresponds to identifying the topics of research articles and the roles of topic items in the context of the article as a whole, using the roles of entities in more local contexts represented by our annotations.

We used the 100-abstract subset of our corpus taken from the corpus used by Gupta and Manning, and the corresponding abstracts in their corpus. Figure 4 shows their original annotation on the same part of the abstract shown in Figure 1, converted to standoff format for displaying in brat. Their annotation is sparser than ours (Figure 1), annotating only terms related to the topic of the paper as a whole. For extraction, they used heuristic rules based on trigger words and Stanford dependencies such as "A term is FOCUS if it is the direct object of the verb *present*" as seed rules. Then, the rule set was enhanced by iteratively adding the head words of extracted phrases as the triggers.

The abstracts were tokenized using the Stanford parser (version 3.4.1) (Klein and Manning 2003), and the tokens are labeled with binary labels for inclusion in Gupta-Manning terms for each topic class (FOCUS, DOMAIN, and TECHNIQUE). Then, the support vector classifier from the python scikit-learn 0.17 package (Pedregosa et al., 2011) with a linear kernel was used to predict the labels.

We tested several combinations of the features from the Stanford parser and our annotation. The features from the Stanford parser were parts of speech (P in Table 5) and the triplet of type, direction (head or argument), and the part of speech of the token it depends/depended on, for each dependency involving the token (D). The features from our annotation were the entity type assigned to the entity mention in which the token is included (T), and the triplet of type, direction, the type of the related entity of the relations that the entity is involved in (R). We also used a location feature for the token: binary features denoting whether the token is in the title, the first, or the last sentence in the abstract text (L).

A binary classifier that determines whether a token belongs to a topic term of the class or not was constructed for each topic class. The class-weight was set to 1:4. We compared the results of "gold" (entities and relations in manual annotation) and "auto" (automatically annotated entities and relations described in the previous section) settings. The syntactic features were in common, derived from the same automatic parsing results.

F1 scores for 10-fold cross validation on the 100 samples are given in Table 5. FOC, DOM, and TEC in the table denotes FOCUS, DOMAIN, and TECHNIQUE. The table also includes the results quoted from (Gupta and Manning 2011), where GM(seed) denotes the results with the seed rules and GM(50) denotes the results with the rules after 50 iterations of enhancement. The result for FOCUS after 50 iteration was not provided in (Gupta and Manning 2011). Note that their results are for their entire set consisting of 474 abstracts.

Although precise comparison is not possible because their count is term-based and ours is token-based, we appear to have achieved results comparable in performance to their rule-based methods with a smaller set of documents. Semantic (T and R) features, combined with the syntactic features, can improve the performance even when automatic annotation results are used. In gold annotation, semantic features alone outperform syntactic features. The results also show that, although relation features contribute to performance more than the entity type feature does (compare T and R), the entity type improves the performance further when combined with relation features, thus showing the positive effect of incorporating entity types.

The contribution of the features is different depending on the topic class. The location feature is effective for FOCUS but not for TECHNIQUE, while semantic features are more effective in finding DOMAIN and TECHNIQUE. In particular, TECHNIQUE can be more effectively found using only semantic features by both automatic and gold annotation than using syntactic features.

The contribution of location feature in finding FOCUS corresponds to the fact that FOCUS (the main topic of the article) is usually stated in the title. In fact, one of the heuristic rules adopted in (Gupta and Manning 2011) was that "if FOCUS is not identified in other rules, let the title be the FOCUS of the article".

The contribution of semantic features in finding TECHNIQUE is related to the nature of TECHNIQUE and DOMAIN: A technology can be a DOMAIN when another technology is applied to achieve it, and can be a TECHNIQUE when it is applied to achieve another technology. Thus, they can be distinguished only in relation to other entities mentioned in the text.

8. Conclusions

We have designed a scheme for annotating entities in computer science/technology domain and the relationship among them using an ontology-based entity typing system and binary relations with role-based relation types. This

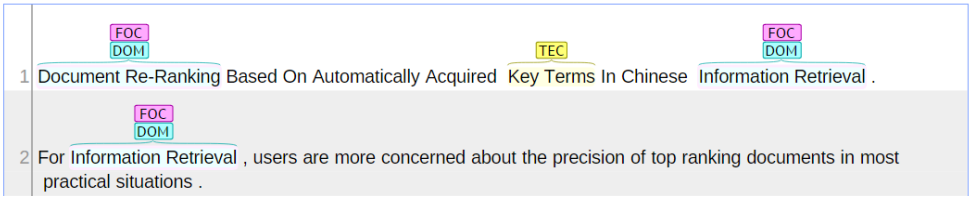

Figure 4: FOCUS-DOMAIN-TECHNIQUE Annotation by Gupta and Manning (2011) on C04-1069

Feature Sets	Gold			Auto		
	FOC	DOM	TEC	FOC	DOM	TEC
T	0.336	0.330	0.358	0.303	0.335	0.321
R	0.353	0.353	0.375	0.301	0.264	0.328
T+R	0.353	0.392	0.383	0.329	0.358	0.327
L+T+R	0.403	0.418	0.387	0.426	0.336	0.328
P+D+T+R	0.439	0.416	0.403	0.415	0.411	0.370
P+D+L+T+R	0.475	0.432	0.403	0.460	0.413	0.374
P+D+L	0.462	0.381	0.319			
GM (seed)	0.553	0.401	0.253			
GM (iter.50)		0.369	0.373			

Table 4: Results for prediction of FOCUS, DOMAIN, TECHNIQUE

scheme enables annotators to annotate the context an entity belongs to in the form of relations to other entities in the same context, representing the role that the entity plays.

We have annotated research abstracts in the ACL anthology and the ACM digital library using our scheme. Although the annotation results are from one expert annotator and the verification of annotation stability is yet to be conducted, we have obtained the following results: (1) the distribution of entity types, although it is a shallow one, can capture the characteristics of the subdomains (ACL vs ACM); (2) the entity type restriction enabled us to find the problems in the current annotation schemes such as the one concerning the relation between the word that denotes an abstract role and the mentions of entities that play the role, and the one in determining the type of mentions with ambiguity/metonymy; (3) the annotation can be used for developing systems for topic extraction from research papers, as the entities and relations annotated using our scheme contributes to distinguish the technology that is the main focus and the technology that is used for achieving it. We are currently refining the scheme on the basis of the results presented in the current paper. The current version of the corpus is available from our Github repository (https://github.com/mynlp/ranis). Development of a search system that incorporates graph-based inference is planned.

9. Bibliographical References

Anick, P., Verhagen, M., and Pustejovsky, J. (2014). Identification of technology terms in patents. *Proceedings of LREC'14*.

Fukuda, S., Nanba, H., and Takezawa, T. (2012). Extraction and visualization of technical trend information from research papers and patents. Proceedings of the 1st International Workshop on Mining Scientific Publications.

Gupta, S. and Manning, C. D. (2011). Analyzing the dynamics of research by Eextracting key aspects of scientific papers. P*roceedings of 5th IJCNLP*.

Kameda, A., Uchiyama, K., Takeda, H., Aizawa, A. (2013). Extraction of Semantic Relationships from Academic Papers using Syntactic Patterns. *Proceedings of eKNOW 2013.*

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

Kim, S. N., Medelyan, O., Kan, M. Y., and Timothy Baldwin. (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. *Proceedings of 41st ACL*.

Miyao, Y., and Tsujii, J. (2008). Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80.

Miwa M. and Sasaki Y. (2014). Modeling joint entity and relation extraction with table representation. *Proceedings of EMNLP'14.*

Nassour-Kassis J., Elhadad M., and Sturm, A. (2015). Building conceptual maps from scientific articles. *Israeli Seminar on Computational Linguistics.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pustejovsky, J., Moszkowicz, J., Batiukova, O., and Rumshisky, A. (2009). GLML: Annotating argument selection and coercion. *Proceedings of the Eight International Conference on Computational Semantics.*

Roth, D. and Yih, W. T., . (2004). A linear programming formulation for global inference in natural language tasks. *Proceedings of Eighth Conference on Computational Natural Language Learning*.

Roth, M., and Klein, E. (2015). Parsing software requirements with an ontology-based semantic role labeler, *Proceedings of the 1st Workshop on Language and Ontologies.*

Roth, M., Diamantopoulos, T., Klein E., Symeonidis, A. (2014). Software requirements: A new domain for semantic parsers, *Proceedings of the ACL 2014 Workshop on Semantic Parsing.*

Sagae, K. and Tsujii, J. (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. *Proceedings of EMNLP-CoNLL'07 shared task*.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsu-jii, J. (2012). brat: a Web-based Tool for NLP-assisted Text Annotation. Proceedings of EACL 2012.

Tateisi, Y., Shidahara, Y. Miyao, Y., Aizawa, A. (2013). Relation Annotation for Understanding Research Papers, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse.*

Tateisi, Y., Shidahara, Y. Miyao, Y., Aizawa, A. (2014). Annotation of computer science papers for semantic relation extraction. *Proceedings of LREC '14.*

10. Acknowledgments

This study was partially supported by the Data Centric Science Research Commons project by Research Organization of Information and Systems, Japan.