Toward an Underspecifiable Corpus Annotation Scheme

Yuka Tateisi
Department of Informatics, Kogakuin University
1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo, 163-8677, Japan
yucca@cc.kogakuin.ac.jp

Abstract

The Wall Street Journal corpora provided for the Workshop on Cross-Framework and Cross-Domain Parser Evaluation Shared Task are investigated in order to see how the structures that are difficult for an annotator of dependency structure are encoded in the different schemes. Non-trivial differences among the schemes are found. The paper also investigates the possibility of merging the information encoded in the different corpora.

1 Background

This paper takes a look at several annotation schemes related to dependency parsing, from the viewpoint of a corpus annotator. The dependency structure is becoming a common criterion for evaluating parsers in biomedical text mining (Clegg and Shepherd, 2007; Pyssalo et al., 2007a), since their purpose in using parsers are to extract predicate-argument relations, which are easier to access from dependency than constituency structure. One obstacle in applying dependency-based evaluation schemes to parsers for biomedical texts is the lack of a manually annotated corpus that serves as a gold-standard. Aforementioned evaluation works used corpora automatically converted to the Stanford dependency scheme (de Marneffe et al., 2006) from gold-standard phrase structure trees in the Penn Treebank (PTB) (Marcus et al., 1993) format. However, the existence of errors in the automatic conversion procedure, which are not well-documented, makes the suitability of the resulting corpus for parser evaluation questionable, especially in comparing PTB-based parsers and parsers based on other formalisms such as CCG and HPSG (Miyao et al., 2007). To overcome the obstacle, we have manually created a dependency-annotated corpus in the biomedical field using the Rasp Grammatical Relations (Briscoe 2006) scheme (Tateisi et al., 2008). In the annotation process, we encountered linguistic phenomena for which it was difficult to decide the appropriate relations to annotate, and that motivated the investigation of the sample corpora provided for the Workshop on Cross-Framework and Cross-Domain Parser Evaluation Shared Task[1], in which the same set of sentences taken from the Wall Street Journal section from Penn Treebank is annotated with different schemes.

The process of corpus annotation is assigning a label from a predefined set to a substring of the text. One of the major problems in the process is the annotator's lack of confidence in deciding which label should be annotated to the particular substring of the text, thus resulting in the inconsistency of annotation. The lack of confidence originates from several reasons, but typical situations can be classified into two types:

1) The annotator can think of two or more ways to annotate the text, and cannot decide which is the best way. In this case, the annotation scheme has more information than the annotator has. For example, the annotation guideline of Penn Treebank (Bies et al. 1995) lists alternatives for annotating structures involving null constituents that exist in the Treebank.

2) The annotator wants to annotate a certain information that cannot be expressed properly with the current scheme. This is to say, the annotator has more information than the scheme can express.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/pe08-st/

Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 17–23
Manchester, August 2008

For example, Tateisi et al (2000) report that, in the early version of the GENIA corpus, some cases of inter-annotator discrepancy occur because the class of names to be assigned (e.g. PROTEIN) is too coarse-grained for annotators, and the result led to a finer-graded classification (e.g. PROTEIN-FAMILY, PROTEIN-COMPLEX) of names in the published version of GENIA (Kim et al., 2003).

In practice, the corpus designers deal with these problems by deciding how to annotate the questionable cases, and describing them in the guidelines, often on an example-by-example basis. Still, these cases are sources of errors when the decision described in the guideline is against the intuition of the annotator.

If the scheme allows the annotator to annotate the exact amount of information that (s)he has, (s)he would not be uncertain about how to annotate the information. However, because the information that an annotator has varies from annotator to annotator it is not practical to define a scheme for each annotator. Moreover, the resulting corpus would not be very useful, for a corpus should describe a "common standard" that is agreed by (almost) everyone.

One solution would be to design a scheme that is as information-rich as possible, in the way that it can be "underspecified" to the amount of the information that an annotator has. When the corpus is published, the annotation can be reduced to the "most-underspecified" level to ensure the uniformity and consistency of annotations, that is, to the level that all the annotators involved can agree (or the corpus can be published as-is with underspecification left to the user). For example, annotators may differ in decision about whether the POS of "human" in the phrase "human annotator" is an NN (common noun) or a JJ (adjective), but everyone would agree that it is not, for example, a VBN (past participle of a verb). In that case, the word can be annotated with an underspecified label like "NN or JJ". The Penn Treebank POS corpus (Santrini, 1990) allows such underspecification (NN|JJ). In the dependency structure annotation, Grammatical Relations (Briscoe 2006), for example, allows underspecification of dependency types by defining the class hierarchy of dependency types. The underspecified annotation is obviously better than discarding the annotation because of inconsistency, for the underspecified annotation have much more information than nothing at all, and can assure consistency over the entire corpus.

Defining an underspecification has another use. There are corpora in similar but different schemes, for a certain linguistic aspect (e.g. syntactic structure) based on formalisms suited for the application that the developers have in mind. That makes the corpus difficult for the use outside the group involved in the development of the corpus. In addition to the difficulty of using the resources across the research groups, the existence of different formalisms is an obstacle for users of NLP systems to compare and evaluate the systems. One scheme may receive a *de facto* status, as is the case with the Penn Treebank, but it is still unsuitable for applications that require the information not encoded in the formalisms or to compare systems based on widely different formalisms (e.g., CCG or HPSG in the case of syntactic parsing).

If some common aspects are extracted from the schemes based on different formalisms, the corpus annotated with the (common) scheme will be used as a standard for (coarse-grained) evaluation and comparison between systems based on different formalisms. If an information-rich scheme can be underspecified into a "common" level, the rich information in the corpus will be used locally for the system development and the "common" information can be used by people outside the developers' group. The key issue for establishing the "common" level would be to provide the systematic way to underspecify the individual scheme.

In this paper, the schemes of dependency corpora provided for the Shared Task are compared on the problematic linguistic phenomena encountered in annotating biomedical abstracts, in order to investigate the possibility of making the "common, underspecified" level of annotation. The compared schemes are mainly CONLL shared task structures (CONLL) [1], Rasp Grammatical Relations (GR) , PARC 700 dependency structures (PARC)[2] and Stanford dependency structures (Stanford; de Marneffe et al. 2006), with partial reference to UTokyo HPSG Treebank predicate-argument structures (HPSG; Miyao 2006) and CCGBank predicate-argument structures (CCG; Hockenmaier and Steedman 2005).

2 Underspecification

In dependency annotation, two types of information are annotated to sentences.

[1] http://www.yr-bcn.es/conll2008/

[2] http://www2.parc.com/isl/groups/nltt/fsbank/triplesdoc.html

- Dependency structure: what is dependent on what
- Dependency type: how the dependent depends on the head

For the latter information, schemes like GR and Stanford incorporates the hierarchy of dependency types and allows systematic underspecification but that does not totally solve the problem. A case of GR is addressed later. If type hierarchy over different schemes can be established, it helps cross-scheme comparison. For the former information, in cases where some information in a corpus is omitted in another (e.g. head percolation), the corpus with less information is considered as the underspecification of the other, but when a different structure is assigned, there is no mechanism to form the underspecified structure so far proposed. In the following section, the sample corpora are investigated trying to find the difference in annotation, especially of the structural difference.

3 How are problematic structures encoded in the sample corpora?

The Wall Street Journal corpora provided for the shared task is investigated in order to look for the structures that the annotator of our dependency corpus commented as difficult, and to see how they are encoded in the different schemes. The subsections describe the non-trivial differences among the annotation schemes that are found. The subsections also discuss the underspecifiable annotation where possible.

3.1 Multi-word Terms

The structure inside multi-word terms, or more broadly, noun-noun sequence in general, have been left unannotated in Penn Treebank, and the later schemes follow the decision. Here, underspecification is realized in practice. In dependency schemes where dependency is encoded by a set of binary relations, the last element of the term is regarded as a head, and the rest of the element of the term is regarded as dependent on the last. In the PARC annotation, proper names like "Los Angeles" and "Alex de Castro" are treated as one token.

However, there are noun sequences in which the head is clearly not the last token. For example, there are a lot of names in the biomedical field where a subtype is specified (e.g. Human Immunodeficiency Virus Type I). If the sequence is considered as a name (of a type of virus in this example), it may be reasonable to assign a flat structure to it, wherever the head is. On the other hand, a flat structure is not adequate for analyzing a structure like "Human Immunodeficiency Virus Type I and Type II". Thus it is conventional to assign to a noun phrase "a flat structure unless coordination is involved" in the biomedical corpora, e.g., GENIA and Bioinfer (Pyssalo et al., 2007b). However, adopting this convention can expose the corpus to a risk that the instances of a same name can be analyzed differently depending on context.

```
Human Immunodeficiency Virus Type
I is a ...
 id(name0, Human Immunodeficiency
 Virus Type I)
 id(name1, Human Immunodeficiency
 Virus)
 id(name2, Type I)
 concat(name0, name1, name2)
 subject(is, name0)

Human Immunodeficiency Virus Type
I and Type II
 id(name3, Type II)
 conj(coord0, name2)
 conj(coord0, name3)
 conj_form(coord0, and)
 adjunct(name1, coord0)
```

Figure 1. PARC-like annotation with explicit annotation of names

A possible solution is to annotate a certain noun sequence as a term with a non-significant internal structure, and where needed, the internal structure may be annotated independently of the outside structure. The PARC annotation can be regarded as doing this kind of annotation by treating a multi-word term as token and totally ignore the internal structure. Going a step further, using IDs to the term and sub-terms, the internal structure of a term can be annotated, and the whole term or a subcomponent can be used outside, retaining the information where the sequence refers to parts of the same name. For example, Figure 1 is a PARC-like annotation using name-IDs, where `id(`ID, `name)` is for assigning an ID to a name or a part of a name, and `name0`, `name1`, `name2`, and `name3` are IDs for "Human Immunodeficiency Virus Type I", "Human Immunodeficiency Virus", "Type I", "Type II", and "Human Immunodeficiency Virus Type II" respectively, and `concat(`*a*, *b*, *c*) means that strings b and c is concatenated to make string *a*.

3.2 Coordination

The example above suggests that the coordination is a problematic structure. In our experience, coordination structures, especially ones with ellipsis, were a major source of annotation inconsistency. In fact, there are significant differences in the annotation of coordination in the sample corpora, as shown in the following subsections.

What is the head?

Among the schemes used in the sample corpora, CCG does not explicitly annotate the coordination but encodes them as if the coordinated constituents exist independently[3]. The remaining schemes may be divided into determination of the head of coordination.

- GR, PARC, and HPSG makes the coordinator (and, etc) the head
- CONLL and Stanford makes the preceding component the head

For example, in the case with "makes and distributes", the former group encodes the relation into two binary relations where "and" is the head (of both), and "makes" and "distributes" are the dependent on "and". In the latter group, CONLL encodes the coordination into two binary relations: one is the relation where "makes" is the head and "and" is the dependant and another where "and" is the head and "distributes" is the dependent. In Stanford scheme, the coordinator is encoded into the type of relation (conj_and) where "makes" is the head and "distributes" is the dependent. As for the CCG scheme, the information that the verbs are coordinated by "and" is totally omitted. The difference of policy on head involves structural discrepancy where underspecification does not seem easy.

Distribution of the dependents

Another difference is in the treatment of dependents on the coordinated head. For example, the first sentence of the corpus can be simplified to "Bell makes and distributes products". The subject and object of the two verbs are shared: "Bell" is the subject of "makes" and "distributes", and "products" is their direct object. The subject is treated as dependent on the coordinator in GR, dependent on the coordinator as well as both verbs in PARC[4], dependent on both verbs in HPSG and Stanford (and CCG), and dependent on "makes" in CONLL. As for the object, "products" is treated as dependent on the coordinator in GR and PARC, dependent on both verbs in HPSG (and CCG), and dependent on "makes" in CONLL and Stanford. The Stanford scheme uniformly treats subject and object differently: The subject is distributed among the coordinated verbs, and the object is treated as dependent on the first verb only.

A different phenomenon was observed for noun modifiers. For example, semantically, "electronic, computer and building products" in the first sentence should be read as "electronic products and computer products and building products" not as "products that have electronic and computer and building nature". That is, the coordination should be read distributively. The distinction between distributive and non-distributive reading is necessary for applications such as information extraction. For example, in the biomedical text, it must be determined whether "CD4+ and CD8+ T cells" denotes "T cells expressing CD4 and T cells expressing CD8" or "T cells expressing both CD4 and CD8".

Coordinated noun modifier is treated differently among the corpora. The coordinated adjectives are dependent on the noun (like in non-distributive reading) in GR, CONLL, and PARC, while the adjectives are treated as separately dependent on the noun in Stanford and HPSG (and CCG). In the PARC scheme, there is a relation named `coord_level` denoting the syntactic type of the coordinated constituents. For example, in the annotation of the first sentence of the sample corpus ("...electronic, computer and building products"), `coord_level(coord~19, AP)` denotes that the coordinated constituents are AP, as syntactically speaking adjectives are coordinated. It seems that distributed and non-distributed readings (semantics) are not distinguished.

It can be said that GR and others are annotating syntactic structure of the dependency while HPSG and others annotate more semantic struc-

[3] Three kinds of files for annotating sentence structures are provided in the original CCGbank corpus: the human-readable corpus files, the machine-readable derivation files, and the predicate-argument structure files.
The coordinators are marked in the human-readable corpus files, but not in the predicate-argument structure files from which the sample corpus for the shared task was derived.

[4] According to one of the reviewers this is an error in the distributed version of the PARC corpus that is the result of the automatic conversion. The correct structure is the one in which the subject is only dependent on both verbs but not on the coordinator (an example is parc_23.102 in http://www2.parc.com/isl/groups/nltt/fsbank/parc700-2006-05-30.fdsc); the same would hold of the object.

ture. Ideally, the mechanism for encoding the syntactic and semantic structure separately on the coordination should be provided, with an option to decide whether one of them is left unannotated.

For example, the second example shown in Figure 1 ("Human Immunodeficiency Virus Type I and Type II") can be viewed as a coordination of two modifiers ("Type I" and "Type II") syntactically, and as a coordination of two names ("Human Immunodeficiency Virus Type I" and "Human Immunodeficiency Virus Type II") semantically. Taking this into consideration, the structure shown in Figure 1 can be enhanced into the one shown in Figure 2 where `conj_sem` is for representing the semantic value of coordination, and `coord0_S` denotes that the dependencies are related semantically to `coord0`. Providing two relations that work as `cood_level` in the PARC scheme, one for the syntactic level and the other for the semantic level, may be another solution: if a parallel of `coord_level`, say, `coord_level_sem`, can be used in addition to encode the semantically coordinated constituents, distributive reading of "electronic, computer and building products" mentioned above may be expressed by `coord_level_sem(coord~19, NP)`indicating that it is a noun phrases with shared head that are coordinated.

```
Human Immunodeficiency Virus Type
I and Type II
 id(name0, Human Immunodeficiency
 Virus Type I)
 id(name1, Human Immunodeficiency
 Virus)
 id(name2, Type I)
 concat(name0, name1, name2)
 id(name3, Type II)
 id(name4, Human Immunodeficiency
 Virus Type II)
 concat(name4, name1, name3)
 conj(coord0, name2)
 conj(coord0, name3)
 conj_form(coord0, and)
 adjunct(name1, coord0)
 conj_sem(coord0_S, name0)
 conj_sem(coord0_S, name4)
```

Figure 2. Annotation of coordinated names on syntactic and semantic levels

Coordinator

Two ways of expressing the coordination between three items are found in the corpora: retaining the surface form or not.

```
cotton , soybeans and rice
eggs and butter and milk
```

For example, the structures for the two phrases above are different in the CONLL corpus while others ignore the fact that the former uses a comma while "and" is used in the latter. That is, the CONLL scheme encodes the surface structure, while others encode the deeper structure, for semantically the comma in the former example means "and". The difference can be captured by retrieving the surface form of the sentences in the corpora that ignore the surface structure. However, encoding surface form and deeper structure would help to capture maximal information and to compare the structures across different annotations more smoothly.

3.3 Prepositional phrases

Another major source of inconsistency involved prepositional phrases. The PP-attachment problem (where the PP should be attached) is a problem traditionally addressed in parsing, but in the case of dependency, the type of attachment also becomes a problem.

Where is the head?

The focus of the PP-attachment problem is the head where the PP should attach. In some cases,a the correct place to attach can be determined from the broader context in which the problematic sentence appears, and in some other cases the attachment ambiguity is "benign" in the sense that there is little or no difference in meaning caused by the difference in the attachment site. However, in highly specialized domain like biomedical papers, annotators of grammatical structures do not always have full access to the meaning, and occasionally, it is not easy to decide where to attach the PP, whether the ambiguity is benign, etc. Yet, it is not always that the annotator of a problematic sentence has no information at all: the annotator cannot usually choose from the few candidates selected by the (partial) understanding of the sentence, and not from all possible sites the PP can syntactically attach.

No schemes provided for the task allow the listing of possible candidates of the phrases where a PP can attach (as allowed in the case of Penn Treebank POS corpus). As with the POS, a scheme for annotating ambiguous attachment should be incorporated. This can be more easily realized for dependency annotation, where the structure of a sentence is decomposed into list of local dependencies, than treebank annotation, where the structure is annotated as a whole. Simply listing the possible dependencies, with a flag for ambiguity, should work for the purpose. Preferably, the flag encodes the information about whether the annotator thinks the ambiguity is benign, i.e. the annotator believes that the ambiguity does not affect the semantics significantly.

Complement or Modifier

In dependency annotation, the annotator must decide whether the PP dependent of a verb or a verbal noun is an obligatory complement or an optional modifier. External resources (e.g. dictionary) can be used for common verbs, but for technical verbs such resources are not yet widely available, and collecting and investigating a large set of actual use of the verbal is not an easy task.

Dependency types for encoding PP-attachment are varied among the schemes. Schemes such as CONLL and Stanford do not distinguish between complements and modifiers, and they just annotate the relation that the phrase "attaches as a PP". HPSG in theory can distinguish complements and modifiers, but in the actual corpus, all PPs appear as modifiers[5]. GR does not mark the type of the non-clausal modifying phrase but distinguish PP-complements (`iobj`), nominal complements (`dobj`) and modifiers. PARC has more distinction of attachment type (e.g. `obj`, `obl`, `adjunct`).

If the inconsistency problem involving the type of PP attachment lies in the distinction between complements and modifiers, treatment of CONLL and Stanford looks better than that of GR and PARC. However, an application may require the distinction (a candidate of such application is relation information extraction using predicate-argument structure) so that analysis with the schemes that cannot annotate such distinction at all is not suitable for such kind of applications. On the other hand, GR does have type-underspecification (Briscoe 2006) but the argument (complement) - modifier distinction is at the top level of the hierarchy and underspecification cannot be done without discarding the information that the dependent is a PP.

A dependent of a verbal has two aspects of distinction: complement/modifier and grammatical category (whether it is an NP, a PP, an AP, etc). The mechanism for encoding these aspects separately should be provided, with an option to decide if one is left unannotated. A possible annotation scheme using IDs is illustrated in Figure 3, where type of dependency and type of the dependent are encoded separately. A slash indicates the alternatives from which to choose one (or more, in ambiguous cases).

```
Dependency(ID, verb, dependent)
Dependent_type(ID, MOD/ARG)
Dependent_form(ID, PP/NP/AP/…)
```

Figure 3: An illustration of attachment to a verbal head

4 Toward a Unified Scheme

The observation suggests that, for difficult lingustic phenomena, different aspects of the phenomena are annotated by different schemes. It also suggests that there are at least two problems in defining the type of dependencies: one is the confusion of the level of analysis, and another is that several aspects of dependency are encoded into one label.

The confusion of the level of analysis means that, as seen in the case of coordination, the syntactic-level analysis and semantic-level analysis receive the same or similar label across the schemes. In each scheme only one level of analysis is provided, but it is not always explicit which level is provided in a particular scheme. Thus, it is inconvenient and annoying for an annotator who wants to annotate the other level or both levels at once.

As seen in the case of PP-dependents of verbals, because different aspects, or features, are encoded in one label, type-underspecification becomes a less convenient mechanism. If labels are properly decomposed into a set of feature values, and a hierarchy of values is provided for each feature, the annotation labels can be more flexible and it is easier for an annotator to choose a label that can encode the desired information. The distinction of syntax/semantics (or there may be more levels) can be incorporated into one of the features. Other possible features include the grammatical categories of head and dependent, argument/modifier distinction, and role of arguments or modifiers like the one annotated in Propbank (Palmer et al., 2005).

Decomposing labels into features have another use. It would make the mapping between one scheme and another more transparent.

As the dependency structure of a sentence is encoded into a list of local information in de-

[5] The modifier becomes a head in HPSG and in CCG unlike other formalisms.

pendency schemes, it can be suggested that taking the union of the annotation of different schemes can achieve the encoding of the union of information that the individual schemes can encode, except for conflicting representations such as the head of coordinated structures, and the head of modifiers in HPSG. If the current labels are decomposed into features, it would enable one to take non-redundant union of information, and mapping from the union to a particular scheme would be more systematic. In many cases listed in the previous section, individual schemes could be obtained by systematically omitting some relations in the union, and common information among the schemes (the structures that all of the schemes concerned can agree) could be retrieved by taking the intersection of annotations. An annotator can annotate the maximal information (s)he knows within the framework of the union, and mapped into the predefined scheme when needed.

Also, providing a mechanism for annotating ambiguity should be provided. As for dependency types the type hierarchy of features described above can help. As for the ambiguity of attachment site and others that involve the problem of what is dependent on what, listing of possible candidates with a flag of ambiguity can help.

Acknowledgments

I am grateful for the anonymous reviewers for suggestions and comments.

References

Bies, Ann, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger , 1995. Bracketing Guidelines for Treebank II Style Penn Treebank Project. Technical report, University of Pennsylvania.

Briscoe, Ted. 2006. An introduction to tag sequence grammars and the RASP system parser. Technical Report (UCAM-CL-TR-662), Cambridge University Computer Laboratory.

Clegg, Andrew B. and Adrian J Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. BMC Bioinformatics 8:24.

Hockenmaier, Julia and Mark Steedman. 2005. CCGbank: User’s Manual, Technical Report (MS-CIS-05-09), University of Pennsylvania.

Kim, J-D., Ohta, T., Teteisi Y., Tsujii, J. (2003). GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics.* 19(suppl. 1), pp. i180-i182.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. Proceedings of LREC 2006, Genoa, Italy.

Miyao, Yusuke. From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model. 2006. PhD Thesis, University of Tokyo.

Miyao, Yusuke, Kenji Sagae, Jun'ichi Tsujii. 2007. Towards Framework-Independent Evaluation of Deep Linguistic Parsers. In Proceedings of Grammar Engineering across Frameworks, Stanford, California, USA, pp. 238-258.

Palmer, Martha, Paul Kingsbury, Daniel Gildea. 2005. "The Proposition Bank: An Annotated Corpus of Semantic Roles". *Computational Linguistics* 31 (1): 71–106.

Pyysalo, Sampo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen, and Tapio Salakoski. 2007a. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. Proceedings of BioNLP Workshop at ACL 2007, Prague, Czech Republic .

Pyysalo, Sampo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen and Tapio Salakoski. 2007b. BioInfer: a corpus for information extraction in the biomedical domain. BMC Bioinformatics 8:50.

Santorini, Beatrice. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. Technical report, University of Pennsylvania.

Tateisi, Yuka, Ohta, Tomoko, Nigel Collier, Chikashi Nobata and Jun'ichi Tsujii. 2000. Building an Annotated Corpus from Biology Research Papers. In the Proceedings of COLING 2000 Workshop on Semantic Annotation and Intelligent Content. Luxembourg. pp. 28-34.

Tateisi,Yuka, Yusuke Miyao, Kenji Sagae, Jun'ichi Tsujii. 2008. GENIA-GR: a Grammatical Relation Corpus for Parser Evaluation in the Biomedical Domain. In the Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, Morocco.