Parallel Entity and Treebank Annotation

Ann Bies
Linguistic Data Consortium
3600 Market Street, 810
Philadelphia, PA 19104
bies@ldc.upenn.edu

Seth Kulick
Institute for Research
in Cognitive Science
3401 Walnut Street
Suite 400A
Philadelphia, PA 19104
skulick@linc.cis.upenn.edu

Mark Mandel
Linguistic Data Consortium
3600 Market Street, 810
Philadelphia, PA 19104
mamandel@ldc.upenn.edu

Abstract

We describe a parallel annotation approach for PubMed abstracts. It includes both entity/relation annotation and a treebank containing syntactic structure, with a goal of mapping entities to constituents in the treebank. Crucial to this approach is a modification of the Penn Treebank guidelines and the characterization of entities as relation components, which allows the integration of the entity annotation with the syntactic structure while retaining the capacity to annotate and extract more complex events.

1 Introduction

A great deal of annotation effort for many different corpora has been devoted to annotation for entities and syntactic structure (treebanks). However, previous efforts at treebanking have largely been independent of the constituency of entities, and previous efforts at entity annotation have likewise been independent of corresponding layers of syntactic structure. We describe here a corpus being developed for biomedical information extraction with levels of both entity annotation and treebank annotation, with a goal that entities can be mapped to constituents in the treebank.

We are collaborating with researchers in the Division of Oncology at The Children's Hospital of Philadelphia, for the purpose of automatically mining the corpus of cancer literature for those associations that link specified variations in individual genes with known malignancies. In particular, we are interested in extracting three entities (Gene, Variation event, and Malignancy) in the following relationship: Gene X with genomic Variation event Y is correlated with Malignancy Z. For example, *WT1 is deleted in Wilms Tumor #5*. In addition, Variation events are themselves relations, consisting of entities representing different aspects of a Variation event.

Mapping entities to treebank constituents is a desirable goal since the entities can then be viewed as semantic types associated with syntactic constituents, and we expect that automated analyses of these related levels will interact in a mutually reinforcing and beneficial way for development of statistical taggers.

In this paper we describe aspects of the entity and treebank annotation that allow this mapping to be largely successful. Potentially large entities that would otherwise cut across syntactic constituents are decomposed into components of a relation. While this is worthwhile by itself on conceptual grounds for entity definition, and was in fact not done for reasons of mapping to syntactic constituents, it makes such a mapping easier. The treebank annotation has been modified from the Penn Treebank guidelines in various ways, such as greater structure for prenominal modifiers. Again, while this would have been done regardless of the mapping of entities, it does make such a mapping more successful.

Previous work on integrating syntactic structure with entity information, as well as relation infor-

Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, pages 21–28,
Ann Arbor, June 2005.

mation, is described in (Miller et al., 2000). Our work is in much the same spirit, although we do not integrate relation annotation into the syntactic trees. PubMed abstracts are quite different from the newswire sources used in that earlier work, with several consequences discussed throughout, such as the use of discontinuous entities.

Section 2 discusses some of the main issues around the development of the guidelines for entity annotation, and Section 3 discusses some of the changes that have been made for the treebank guidelines. Section 4 describes the annotation workflow and the resulting merged representation. Section 5 evaluates the mapping between entities and constituents, and Section 6 is the conclusion.

2 Guidelines for Entity Annotation

Here we give a summary of the main features of our annotation guidelines. We have been influenced in this by the annotation guidelines for the Automatic Content Extraction (ACE) project (Consortium, 2004).[1] However, our source materials are medical abstracts from PubMed[2], and important differences between the domains have required significant changes and additions to many definitions, guidelines, and procedures.

Most obviously, the vocabulary is very different. Many of the tokens in our source texts are chemical terms with a complex productive morphology, and a certain number are unique in PubMed. Many others are strings of notation, like *S37F*, often containing relevant entity references that must be isolated (*S*, *37*, and *F*). And even apart from these, we are looking at a very different dialect of English from that used by the Wall Street Journal and the Associated Press. Annotation of English newswire requires native English competency; entity annotation of biomedical English requires a background in biology as well.

The entity instances in the text are also qualitatively different. Instead of individual pieces of the physical or social universe – *Emanuel Sosa, the Eiffel Tower, the man in the yellow hat* – we have abstractions, categories that are not to be confused with their instantiations: *neuroblastoma*, *K-ras* (a gene), *codon 42*.[3] We are not currently annotating pronominal or other forms of coreference.

2.1 Entities Annotated

2.1.1 Gene Entity

For the sake of this project the definition for "Gene Entity" has two significant characteristics. First, as just mentioned, "Gene" refers to a conceptual entity as opposed to the specific manifestation of a gene (e.g., not the "K-ras" in some specific cell in some individual, but an abstraction that cannot be pointed to).

Second, "Gene" refers to a composite entity as opposed to the strict biological definition. There are often ambiguities in the usage of the entity names. I is sometimes unclear as to whether the gene or protein is being referenced, and the same name can refer to the gene or the protein at different locations in the same document. In a similar way as the ACE project allows "geopolitical" entities to have different roles, such as "location" or "organization", we consider a "Gene" to be a composite entity that can have different roles throughout a document. Therefore, Gene entity mentions can have types Gene-generic, Gene-protein, and Gene-RNA.

2.1.2 Variation Events as Relations

As mentioned in the introduction, Variation events are relations between entities representing different aspects of a Variation; specifically, a Variation is a relationship between two or more of the following entities: Type (e.g., *point mutation*, *translocation*, or *inversion*), Location (e.g., *codon 14*, *1p36.1*, or *base pair 278*), Original-State and Altered-State (e.g., *Thymine*).

The entities as such are independent and unconnected. We add a level of *relation* to annotate the associations between them: For example, the text fragment *a single nucleotide substitution at codon 249, predicting a serine to cysteine amino acid substitution (S249C)* contains the entities:

Variation-type substitution

Variation-location codon 249

[1] Another source of influence is previous work in annotation for biomedical information extraction, such as (Ohta et al., 2002). Space prevents adequate discussion of here of the differences.

[2] http://www.ncbi.nlm.nih.gov/entrez/

[3] This domain shows no such clear distinction between Name and Nominal mentions as in the texts covered by ACE.

Variation-state-original serine

Variation-state-altered cysteine

These entities are annotated individually but are also collected into a single Variation relation.

It is also possible for a Variation relation to arise from a more compact collection of entities. For example, the text *S249C* consists of three entities collected into a Variation relation:

Variation-location 249

Variation-state-original S

Variation-state-altered C

These four components represent the key elements necessary to describe any genomic variation event. Variations are often underspecified in the literature. For example, the first relation above has all four components while the second is missing the Variation-type. Characterizing individual Variations as relations among such components provides us with a great deal of flexibility.

The "Gene" entities are analogous to the ACE geopolitical entity, in that the second part of the entity names ("-RNA", "-generic","-protein") disambiguates the metonymy of the "Gene". The subtypes of the Variation entities, in contrast, indicate different kinds of entities in their own right, which can also function as components of a Variation relation.

2.1.3 Malignancy

The Malignancy annotation guidelines were under development during the annotation of the corpus described here. While they have since been more completely defined, they are not included as part of the annotated files discussed here, and so are not further discussed in this paper.

2.2 Discontinuous Entities

We have introduced a mechanism we call "chaining" to annotate discontinuous entities, which may be more common in abstracts than in full text because of the pressure to reduce word count. For example, in *K- and N-ras* there are two entities, *K-ras* and *N-ras*, of which only the second is a solid block of text. Our entity annotators are allowed to change the tokenization if necessary to isolate the components of *K-ras*:

text `K- and N-ras`

original tokenization `[K-][and][N-ras]`

		Multiple Tokens	
Entity Type	Single Tokens	Non-chains	Chains
Gene-generic	104	6	0
Gene-protein	921	349	6
Gene-RNA	1987	156	36
Var-location	95	445	125
Var-state-orig	151	5	0
Var-state-altered	162	10	0
Var-type	235	271	1

Table 1: Entity Instances

modified tokenization

`[K][-][and][N][-][ras]`

entity annotation

1. `K- ... ras` (chain with separated tokens)
2. `N-ras` (contiguous tokens)

2.3 Entity Frequencies

Table 1 shows the number of instances of each of the entity types in the 318 abstracts, discussed further in Section 4, that have been both entity annotated and treebanked. We separate the entities into single-token and multiple-token categories since it is only the multiple-token categories that raise an issue for mapping constituents.

3 Treebank Annotation

The Penn Treebank II guidelines (Bies et al., 1995) were followed as closely as possible, but the nature of the biomedical corpus has made some changes necessary or desirable. We have also taken this opportunity to address several long-standing issues with the original set of guidelines, with regard to NP structure in particular. This has resulted in the introduction of one new node label for sub-NP nominal substrings (NML). One additional empty category (*P*) has been introduced in order to improve the match-up of chained entity categories with treebank nodes. It is used as a placeholder to represent distributed modification in nominals and does not represent the trace of movement.

3.1 Tokenization/Part-of-Speech

We have also adopted several changes in word-level tokenization, leading to a number of part-of-speech and structural differences as well. Many hyphenated words are now treated as separate tokens (*New York - based* would be four tokens, for example). These hyphens now have the part-of-speech tag HYPH. If the separated prefix is a morphological unit that does not exist as a free-standing word, it has the part-of-speech tag AFX. With chemical names and scientific notation in the biomedical corpus in particular, spaces and punctuation may occur within a single "token", which will have a single POS tag.

3.2 Right-Branching Default

We assume a default binary right-branching structure under any NP and NML node. Each daughter of the phrase (whether a single token or itself a constituent node) is assumed to have scope over everything to its right. This means that every daughter also forms a constituent with everything to its right. This assumption makes the annotation process for multi-token nominals less complex and the resulting trees more legible, but still allows us to readily derive constituent nodes not explicitly represented. For example, in

```
(NP (JJ primary) (NN liver)
    (NN cancer))
```

we assume that "liver cancer" is a constituent, and that "primary" has scope over it.

So, although we do not show the intermediate nodes explicitly in our annotation, our assumed structure for this NP could be derived as

```
(NP (JJ primary)
    (newnode (NN liver)
          (newnode (NN cancer))))
```

As discussed in Section 5, entities sometimes map to such implicit constituents, and a node needs to be added to make the constituent explicit so the the entity can be mapped to it.

3.3 New Node Level for Non-Right-Branching: NML

We use the NML node label to mark nominal subconstituents that do not follow the default binary right-branching structure. Any two or more non-final elements that form a constituent are bound together by NML.

```
(NP (NML (NN human)
         (NN liver)
         (NN tumor))
    (NN analysis))
```

3.4 New Empty Category for Distributed Readings within NP: *P*

As discussed in Section 2.2, discontinuous entities are annotated using the "chaining" mechanism. Analogously, we have introduced a placeholder, *P*, for distributed material in the treebank. It is used exclusively in coordinated nominal structures, placed in coordinated elements that are missing either a distributed head or a distributed premodifier. In *K- and N-ras*, the coordinated premodifier *K-* is missing the distributed head *ras*, so the placeholder *P* is inserted after *K-* and coindexed with *ras*:

```
(NP (NP (NN K) (HYPH -)
        (NML-1 (-NONE- *P*)))
    (CC and)
    (NP (NN N) (HYPH - )
        (NML-1 (NN ras))))
```

This creates constituent nodes *K-ras* and *N-ras* that align with the entities being represented by chaining.[4]

4 Annotation Process

The annotation process comprises the following steps: Paragraph and sentence annotation (including the delimitation of irrelevant text such as author names); tokenization; entity annotation; part-of-speech (POS) annotation; treebanking; merged representation.

Entity annotation precedes POS annotation, since the entity annotators often have to correct the tokenization, which affects the POS labels. For example, *nephro- and hepatocarcinoma* refers to two entities, *nephrocarcinoma* and *hepatocarcinoma*, and so the entity annotator would split *hepatocarcinoma* into two tokens, for chaining *nephro* and *carcinoma*

[4] In spite of the apparent similarity between *P* and right node raising structures (*RNR*), they are not interchangeable as the shared element often occurs to the left rather than the right (e.g., *codon 12 or 13* in Section 5.3).

(see Section 2.2). Since the entity annotators are not qualified for POS annotation, doing POS annotation after entity annotation allows the POS annotators to annotate any such tokenization changes.

Treebank annotation uses the same tokenization as for the corresponding entity file. Continuing the above example, the treebank file would have separate tokens for *hepato* and *carcinoma*. Note that this would be the case even if we did not have the goal of mapping entities to constituents. It arises from the more minimal requirement of maintaining identical tokenization in the treebank and entity files, and so leads to changes in treebank annotation such as discussed in Section 3.4.

All of the annotation steps except entity annotation use automated taggers (or a parser in the case of treebanking),[5] producing annotation that then gets hand-corrected.

The use of the parser for producing a parse for correction by the treebankers include a somewhat unusual feature that arises from our parallel entity and treebank annotation. The parser that we are using, (Bikel, 2004),[6] allows prebracketing of parts of the parser input, so that the parser will respect the prebracketing. We use this ability to prebracket entities, which can also help to disambiguate the constituencies for prenominal modifiers, which can often be unclear for annotators without a medical background. For example, the input to the parser might contain something like:

```
...(NN activation)
   (IN of)
   (PRP$ its)
   (* (NN tyrosine)
      (NN kinase) )
   (NN activity)...
```

indicating by the `(* )` that *tyrosine kinase* should be a constituent. (It is a Gene-protein.)

Our first release of data, PennBioIE Release 0.9 (`http://bioie.ldc.upenn.edu/publications`), contains 1157 oncology PubMed abstracts, all annotated for entities and POS, of which 318 have also been treebanked. The website also contains full documentation for the various annotation guidelines mentioned in this paper.

[5] Entity taggers have been developed (McDonald et al., 2004) but have not yet been integrated into the project.

[6] Available at `http://www.cis.upenn.edu/~dbikel/software.html#stat-parser`

```
;sentence 4 Span:331..605
;In the present study, we screened for
;the K-ras exon 2 point mutations in a
;group of 87 gynecological neoplasms
;(82 endometrial carcinomas, four
;carcinomas  of the uterine cervix and
;one uterine carcinosarcoma) using the
;non-isotopic  PCR-SSCP-direct
;sequencing techniques.
;[373..378]:gene-rna:"K-ras"
;[379..385]:variation-location:"exon 2"
;[386..401]:variation-type:
            "point mutations"
(SENT
  (S
    (PP (IN:[331..333] In)
      (NP (DT:[334..337] the)
          (JJ:[338..345] present)
          (NN:[346..351] study)))
    (,:[351..352] ,)
    (NP-SBJ (PRP:[353..355] we))
    (VP (VBD:[356..364] screened)
      (PP-CLR (IN:[365..368] for)
        (NP (DT:[369..372] the)
            (NN:[373..378] K-ras)
            (NML (NN:[379..383] exon)
                 (CD:[384..385] 2))
            (NN:[386..391] point)
            (NNS:[392..401] mutations)))
      (PP (IN:[402..404] in)
        (NP
          (NP (DT:[405..406] a)
              (NN:[408..413] group))
          (PP (IN:[414..416] of)
            (NP (CD:[417..419] 87)
                (JJ:[420..433]
                    gynecological)
                (NNS:[434..443]
                     neoplasms)
[...]
```

Figure 1: Example .mrg file

4.1 Example of Merged Output

The 318 files that have been both treebanked and entity annotated are also available in a merged "`.mrg`" format. The treebank and entity annotations are both stand-off, referring to character spans in the same source file, and we take advantage of this so that the merged representation relates the entities and constituents by these spans. Figure 1 shows a fragment of one such `.mrg` file.

This `.mrg` file excerpt shows the text of sentence 4 in the file, which spans the character offsets 331..605. Each entity is listed by span (which can in-

clude several tokens), entity type, and the text of the entity. The treebank part is the same basic format as the `.mrg` files from the Penn Treebank, except that each terminal has the format

```
(POSTag:[from..to] terminal)
```

where `[from..to]` is that terminal's span in the source file.

The first entity listed, *K-ras*, is a Gene-RNA entity with span `[373..378]`, which corresponds to the single token:

```
(NN:[373..378] K-ras)
```

The second entity, *exon 2*, is a Variation-location with span `[379..385]`, which corresponds to the two tokens:

```
(NN:[379..383] exon)
(CD:[384..385] 2)
```

The third entity, *point mutations*, is a Variation-type with span `[386..401]`, which corresponds to the two tokens:

```
(NN:[386..391] point)
(NNS:[392..401] mutations)
```

By including the terminal span information in the treebank, we make explicit how the tokens that make up the entities are treated in the treebank representation.

5 Entity-Constituent Mapping

One of our goals for the release of the corpus is to allow users to choose how they wish to handle the integration of the entity and treebank information. By providing the corresponding spans for both aspects of the annotation, we provide the raw material for any integrated approach.

We therefore do not attempt to force the entities and constituents to line up perfectly. However, given the parallel annotation just illustrated, we can analyze how close we come to the ideal of the entities behaving as semantic types on syntactic constituents.

5.1 Mapping Categories

Leaving aside chains for the moment, we categorize each entity/treebank mapping in one of three ways:

Exact match There is a node in the tree that yields exactly the entity. For example, the entity *exon 2* in Figure 1

```
;[379..385]:variation-location:
            "exon 2"
```

corresponds exactly to the `NML` node in Figure 1

```
(NML (NN:[379..383] exon)
     (CD:[384..385] 2))
```

Missing node There is no node in the tree that yields exactly that entity, but it is possible to add a node to the tree that would yield the entity. A common reason for this is that the default right branching treebank annotation (Section 3.2) does not make explicit the required node.

For example, the entity *point mutations* in Figure 1

```
;[386..401]:variation-type:
            "point mutations"
```

does not correspond to a node in the relevant part of the tree:

```
(NP (DT:[369..372] the)
    (NN:[373..378] K-ras)
    (NML (NN:[379..383] exon)
         (CD:[384..385] 2))
    (NN:[386..391] point)
    (NNS:[392..401] mutations))
```

However, it is possible to insert a node into the tree to yield exactly the entity:

```
(NP (DT:[369..372] the)
    (NN:[373..378] K-ras)
    (NML (NN:[379..383] exon)
         (CD:[384..385] 2))
    (newnode (NN:[386..391] point)
             (NNS:[392..401]
                    mutations)))
```

Note that this node corresponds exactly to the implicit constituency assumed by the right branching rule. For our own internal research purposes we have generated a version of the treebank with such nodes added, although they are not in the current release.

Crossing The most troublesome case, in which the entity does not match a node in the tree and also cuts across constituent boundaries, so it is not even possible to add a node yielding the entity. Typically this

Entity Type	Total	Exact Match	Miss-ing	Cross-ing
Gene-generic	6	4	1	1
Gene-protein	349	236	103	10
Gene-RNA	156	115	35	6
Var-location	445	348	68	29
Var-state-orig	5	3	1	1
Var-state-altered	10	8	0	2
Var-type	271	123	142	6
Total	1242	837	350	55

Table 2: Matching Status of Non-Chained Multiple Token Instances

Entity Type	Total	Exact Match	Not Exact Match
Gene-generic	0	0	0
Gene-protein	6	4	2
Gene-RNA	36	29	7
Var-location	125	103	22
Var-state-orig	0	0	0
Var-state-altered	0	0	0
Var-type	1	0	1
Total	168	136	32

Table 3: Matching Status of Chained Multiple Token Instances

is due to an entity containing text corresponding to a prepositional phrase. For example, the sentence

```
One ER showed a G-to-T mutation in
the second position of codon 12
```

has the entity

```
[1280..1307]:variation-location:
            "second position
             of codon 12"
```

The relevant part of the corresponding tree is

```
(PP-LOC (IN:[1272..1274] in)
  (NP
    (NP (DT:[1276..1279] the)
        (JJ:[1280..1286] second)
        (NN:[1287..1295] position))
    (PP (IN:[1296..1298] of)
        (NP (NN:[1299..1304] codon)
            (CD:[1305..1307] 12)))))
```

Due to the inclusion of the determiner in the NP *the second position*, while it is absent from the entity definition which does include the following PP, it is not possible to add a node to the tree yielding exactly *second position of codon 12*.[7] It is possible to relax the requirements on exact match to include the determiner.[8]

However, one of our initial goals in this investigation was to determine whether this sort of limited crossing is indeed a major source of the mapping mismatches.

5.2 Overall Mapping Results

Table 2 is a breakdown of how well the (non-chain) entities can be mapped to constituents. Here we are concerned only with entities that consist of multiple tokens, since single-token entities can of course map directly to the relevant token.

The number of crossing cases is relatively small. One reason for this is the use of relations for breaking potentially large entities into component parts, since the component entities either already map to an entity or can easily be made to do so by making implicit constituents explicit to disambiguate the tree structure. The crossing cases tend to be ones in which the entities are in a sense a bit too "big", such as including a prepositional phrase.[9]

[7] The inclusion of the PP in an entity can be a problem for the constituent mapping even aside from the determiner issue. It is possible for the PP, such as *of codon 12*, to be followed by another PP, such as *in K-ras*. Since all PPs are attached at the same level, *of codon 12* and *in K-ras* are sisters, and so, even if the determiner was included in the entity name, there is no constituent consisting of just *the second position of codon 12*. However, in that case it is then possible to add a node yielding the NP and first PP. A similar issue sometimes arises when attempting to relate Propbank arguments to tree constituents.

[8] Another alternative would be to modify the treatment of noun phrases and determiners in the treebank annotation to be more akin to DPs. However, this has proved to be an impractical addition to the annotation process.

[9] As discussed in Section 4, we are prebracketing entities in the parses prepared for the treebankers to correct. There are two possibilities for how the entities can therefore ever cross treebank constituents: (1) the treebank annotation was done before we started doing such prebracketing, so the treebank annotator was not aware of the entities, or (2) the prebracketing was in-

5.3 Chained Entities

Table 3 shows the matching status of multiple token instances that are also chains (and so were not included in Table 2). The presence of chains is mostly localized to certain entity types, and the mapping is mostly successful. Variation-location contains many of the chains due to the occurrences of phrases such as *codon 12 or 13*, which map exactly to the corresponding use of the `*P*` placeholder, such as:

```
(NP (NP
      (NML-1 (NN codon))
      (CD 12))
    (CC or)
    (NP
      (NML-1 (-NONE- *P*))
      (CD 13)))
```

Cases that do not map exactly are ones in which the syntactic context does not permit the use of the placeholder *P*. For example, the text *specific codons (12, 13, and 61)*, has three discontinuous entities (*codons..12*, *codons..13*, *codons..61*), but the parenthetical context does not permit using the placeholder `*P*`:

```
(NP (JJ specific) (NNS codons)
  (PRN (-LRB- -LRB-)
    (NP (NP (CD 12))
        (, ,)
        (NP (CD 13))
        (, ,) (CC and)
        (NP (CD 61)))
    (-RRB- -RRB-)))
```

and so this example contains three mismatches.

6 Conclusion

We have described here parallel syntactic and entity annotation and how changes in the guidelines facilitate a mapping between entities and syntactic constituents. Our main purpose in this paper has been to investigate the success of this mapping. As Tables 2 and 3 show, once we make explicit the implicit right-branching binary structure, only 6.2%[10] of the entities cannot be mapped directly to a node in the tree. It also appears likely that a significant percentage of even the non-matching cases can match as well, with a slight relaxation of the matching requirement (e.g., allowing entities to have an optional determiner).

We view this in part as a successful experiment illustrating how both linguistic content and entity annotation can be enhanced by their interaction. We expect this enhancement to be useful both for biomedical information extraction in particular and more generally for the development of statistical systems that can take into account different levels of annotation in a mutually beneficial way.

Acknowledgements

The project described in this paper is based at the Institute for Research in Cognitive Science at the University of Pennsylvania and is supported by grant EIA-0205448 from the National Science Foundation's Information Technology Research (ITR) program. We would like to thank Yang Jin, Mark Liberman, Eric Pancoast, Colin Warner, Peter White, and Scott Winters for their comments and assistance, as well as the invaluable feedback of all the annotators listed at `http://bioie.ldc.upenn.edu/index.jsp?page=aboutus.html`.

References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II Style, Penn Treebank Project. Tech report MS-CIS-95-06, University of Pennsylvania, Philadelphia, PA.

Daniel M. Bikel. 2004. *On the Parameter Space of Lexicalized Statistical Parsing Models*. Ph.D. thesis, Department of Computer and Information Sciences, University of Pennsylvania.

Linguistic Data Consortium. 2004. Annotation guidelines for entity detection and tracking (edt), version 4.2.6 200400401. http://www.ldc.upenn.edu/Projects/ACE/docs/EnglishEDTV4-2-6.PDF.

Ryan McDonald, Scott Winters, Mark Mandel, Yang Jin, Pete White, and Fernando Pereira. 2004. An entity tagger for recognizing acquired genomic variations in cancer literature. *Bioinformatics*, 22(20):3249–3251.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *6th Applied Natural Language Processing Conference*.

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, and Jun'ici Tsuji. 2002. The GENIA corpus: An annotated corpus in molecular biology domain. In *Proceedings of the 10th International Conference on Intelligent Systems for Molecular Biology*.

deed done, but the treebank annotator could not abide by the resulting tree and modified the parser output accordingly.

[10] 1410 total multiple token entities, both chained and non-chained, with 87 cases that cannot be mapped (55 crossing, 32 chained non-exact match).