Coling 2008

22nd International Conference on Computational Linguistics

Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation

Workshop chairs:

Johan Bos, Edward Briscoe, Aoife Cahill, John Carroll, Stephen Clark, Ann Copestake, Dan Flickinger, Josef van Genabith, Julia Hockenmaier, Aravind Joshi, Ronald Kaplan, Tracy Holloway King, Sandra Kübler, Dekang Lin, Jan Tore Lønning, Christopher Manning, Yusuke Miyao, Joakim Nivre, Stephan Oepen, Kenji Sagae, Nianwen Xue, and Yi Zhang

23 August 2008

Order copies of this and other Coling proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Introduction

Broad-coverage parsing has come to a point where distinct approaches can offer (seemingly) comparable performance: statistical parsers acquired from the Penn Treebank (PTB); data-driven dependency parsers; ‘deep’ parsers trained off enriched treebanks (in linguistic frameworks like CCG, HPSG, or LFG); and hybrid ‘deep’ parsers, employing hand-built grammars in, for example, HPSG, LFG, or LTAG. Evaluation against trees in the Wall Street Journal (WSJ) section of the PTB has helped advance parsing research over the course of the past decade. Despite some scepticism, the crisp and, over time, stable task of maximizing ParsEval metrics (i.e. constituent labeling precision and recall) over PTB trees has served as a dominating benchmark. However, modern treebank parsers still restrict themselves to only a subset of PTB annotation; there is reason to worry about the idiosyncrasies of this particular corpus; it remains unknown how much the ParsEval metric (or any intrinsic evaluation) can inform NLP application developers; and PTB-style analyses leave a lot to be desired in terms of linguistic information.

The Grammatical Relations (GR) scheme, inspired by Dependency Grammar, offers a level of abstraction over specific syntactic analyses. It aims to capture the ‘gist’ of grammatical relations in a fashion that avoids reference to a token linguistic theory. GR has recently been applied successfully in a series of cross-framework parser evaluation studies. At the same time, rather little GR gold standard data is available, and the GR scheme has been questioned for some of its design decisions. More specifically, GR builds on a combination of syntactic and, albeit very limited, some semantic information. Existing studies suggest that the GR gold standard can be both overly rich and overly shallow in some respects. Furthermore, the mapping of ‘native’ parser outputs into GR introduces noise, and it raises a number of theoretical and practical questions.

Gold standard representations at the level of propositional semantics have at times been proposed for cross-framework parser evaluation, specifically where the parsing task is broadly construed as a tool towards ‘text understanding’, i.e. where the parser is to provide all information that is grammaticalized and contributing to interpretation. PropBank would seem a candidate gold standard, but to date very few studies exist that report on the use of PropBank for parser evaluation. The reasons might be that (at least some) parser developers believe that PropBank goes too far beyond the grammatical level to serve for parser evaluation, and that starting from PTB structures may have led to some questionable annotation decisions.

Finally, a complementary topic to cross-framework evaluation is the increasing demand for cross-domain parser evaluation. At conferences in 2007, concerns were expressed about results that might rely on particular properties of the WSJ PTB, and over idiosyncrasies of this specific sample of natural language. For example, it remains a largely open question to what degree progress made in PTB parsing can carry over to other genres and domains; a related question is on the fitness of some specific approach (when measured in parser evaluation metrics) for actual NLP applications. In summary, it may be necessary that the WSJ- and PTB-derived parser benchmarks be complemented by other gold standards, both in terms of the selection of texts and target representations. And to further the adaptation of parser evaluation to more languages, it will be important to carefully distill community experience from ParsEval and GR evaluations.

This workshop aims to bring together developers of broad-coverage parsers who are interested in questions of target representations and cross-framework and cross-domain evaluation and benchmarking. From informal discussions that the co-organizers had among themselves and with colleagues, it seems evident that there is comparatively broad awareness of current issues in parser evaluation, and a lively interest in detailed exchange of experience (and beliefs). Specifically, the organizers have tried to attract representatives from diverse parsing approaches and frameworks, ranging from 'traditional' treebank parsing, over data-driven dependency parsing, to parsing in specific linguistic frameworks. For the latter class of parsers, in many frameworks there is a further sub-division into groups pursuing 'classic' grammar engineering vs. ones who rely on grammar acquisition from annotated corpora.

Quite likely for the first time in the history of these approaches, there now exist large, broad-coverage parsing systems representing diverse traditions that can be applied to running text, often producing comparable representations. In our view, these recent developments present a new opportunity for re-energizing parser evaluation research. We sincerely wish this workshop will provide participants with the opportunity for in-depth and cross-framework exchange of expertise and discussion of future directions in parser evaluation.

A specific sub-goal of the workshop is to establish an improved shared knowledge among participants of the strengths and weaknesses of extant annotation and evaluation schemes. In order to create a joint focus for detailed discussion, the workshop preparation included a 'lightweight' shared task. For a selection of 50 sentences (of which ten were considered obligatory, the rest optional) for which PTB, GR, and PropBank (and other) annotations are available, contributors were invited to scrutinize existing gold-standard representations contrastively, identify perceived deficiencies, and sketch what can be done to address these. As an optional component, participants in the shared task were welcome to include 'native', framework-specific output representations and actual results for a parsing system of their choice (be it their own or not) in the contrastive study. In either case, submissions to the shared task reflect on the nature of different representations, highlight which additional distinctions are made in either scheme, and argue why these are useful (for some task) or unmotivated (in general). Of the eight papers selected for presentation at the workshop, the following three were submissions to the shared task, viz. those by Flickinger (page 17), Tateisi (page 24), and McConville and Dzikovska (page 51). For further information on the workshop as a whole, its shared task, and some specific datasets used, please see:

`http://lingo.stanford.edu/events/08/pe/`

Organizers:

Johan Bos, University of Rome ‘La Sapienza’ (Italy)
Edward Briscoe, University of Cambridge (UK)
Aoife Cahill, University of Stuttgart (Germany)
John Carroll, University of Sussex (UK)
Stephen Clark, Oxford University (UK)
Ann Copestake, University of Cambridge (UK)
Dan Flickinger, Stanford University (USA)
Josef van Genabith, Dublin City University (Ireland)
Julia Hockenmaier, University of Illinois at Urbana-Champaign (USA)
Aravind Joshi, University of Pennsylvania (USA)
Ronald Kaplan, Powerset, Inc. (USA)
Tracy Holloway King, PARC (USA)
Sandra Kübler, Indiana University (USA)
Dekang Lin, Google Inc. (USA)
Jan Tore Lønning, University of Oslo (Norway)
Christopher Manning, Stanford University (USA)
Yusuke Miyao, University of Tokyo (Japan)
Joakim Nivre, Växjö and Uppsala Universities (Sweden)
Stephan Oepen, University of Oslo (Norway) and CSLI Stanford (USA)
Kenji Sagae, University of Southern California (USA)
Nianwen Xue, University of Colorado (USA)
Yi Zhang, DFKI GmbH and Saarland University (Germany)

Table of Contents

Conference Programme

Saturday, August 23, 2008

	Session 1 (Chair: Ann Copestake)
9:00–9:30	Workshop Motivation and Overview (Cahill, Oepen, et al.)
9:30–10:00	*The Stanford Typed Dependencies Representation* Marie-Catherine de Marneffe and Christopher D. Manning
10:00–10:30	*Exploring an Auxiliary Distribution Based Approach to Domain Adaptation of a Syntactic Disambiguation Model* Barbara Plank and Gertjan van Noord
10:30–11:00	Coffee Break
	Session 2 (Chair: John Carroll)
11:00–11:30	*Toward an Underspecifiable Corpus Annotation Scheme* Yuka Tateisi
11:30–12:00	*Toward a Cross-Framework Parser Annotation Standard* Dan Flickinger
12:00–12:30	Discussion
12:30–14:00	Lunch Break

Saturday, August 23, 2008 (continued)

Session 3 (Chair: Kenji Sagae)

14:00–14:30 Summary of CoNLL 2008 Shared Task (Nivre)

14:30–15:00 *Parser Evaluation Across Frameworks without Format Conversion*
Wai Lok Tam, Yo Sato, Yusuke Miyao and Junichi Tsujii

15:00–15:30 *Large Scale Production of Syntactic Annotations to Move Forward*
Anne Vilnat, Gil Francopoulo, Olivier Hamon, Sylvain Loiseau, Patrick Paroubek and Eric Villemonte de la Clergerie

15:30–16:00 Coffee Break

Session 4 (Chair: Tracy Holloway King)

16:00–16:30 *Constructing a Parser Evaluation Scheme*
Laura Rimell and Stephen Clark

16:30–17:00 *'Deep' Grammatical Relations for Semantic Interpretation*
Mark McConville and Myroslava O. Dzikovska

17:00–17:30 Discussion

The Stanford typed dependencies representation

Marie-Catherine de Marneffe
Linguistics Department
Stanford University
Stanford, CA 94305
mcdm@stanford.edu

Christopher D. Manning
Computer Science Department
Stanford University
Stanford, CA 94305
manning@stanford.edu

Abstract

This paper examines the Stanford typed dependencies representation, which was designed to provide a straightforward description of grammatical relations for any user who could benefit from automatic text understanding. For such purposes, we argue that dependency schemes must follow a simple design and provide semantically contentful information, as well as offer an automatic procedure to extract the relations. We consider the underlying design principles of the Stanford scheme from this perspective, and compare it to the GR and PARC representations. Finally, we address the question of the suitability of the Stanford scheme for parser evaluation.

1 Introduction

The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relationships in a sentence that could easily be understood and effectively used by people without linguistic expertise who wanted to extract textual relations. The representation was not designed for the purpose of parser evaluation. Nevertheless, we agree with the widespread sentiment that dependency-based evaluation of parsers avoids many of the problems of the traditional Parseval measures (Black et al., 1991), and to the extent that the Stanford dependency representation is an effective representation for the tasks envisioned, it is perhaps closer to an appropriate task-based evaluation than some of the alternative dependency representations available. In this paper we examine the representation and its underlying design principles, look at how this representation compares with other dependency representations in ways that reflect the design principles, and consider its suitability for parser evaluation.

A major problem for the natural language processing (NLP) community is how to make the very impressive and practical technology which has been developed over the last two decades approachable to and usable by everyone who has text understanding needs. That is, usable not only by computational linguists, but also by the computer science community more generally and by all sorts of information professionals including biologists, medical researchers, political scientists, law firms, business and market analysts, etc. Thinking about this issue, we were struck by two facts. First, we noted how frequently WordNet (Fellbaum, 1998) gets used compared to other resources, such as FrameNet (Fillmore et al., 2003) or the Penn Treebank (Marcus et al., 1993). We believe that much of the explanation for this fact lies in the difference of complexity of the representation used by the resources. It is easy for users not necessarily versed in linguistics to see how to use and to get value from the straightforward structure of WordNet. Second, we noted the widespread use of MiniPar (Lin, 1998) and the Link Parser (Sleator and Temperley, 1993). This clearly shows that (i) it is very easy for a non-linguist thinking in relation extraction terms to see how to make use of a dependency representation (whereas a phrase structure representation seems much more foreign and forbidding), and (ii) the availability of high quality, easy-to-use (and preferably free) tools is essential for driving broader use of NLP tools.[1]

[1] On the other hand, evaluation seems less important; to the best of our knowledge there has never been a convincing and thorough evaluation of either MiniPar or the Link Grammar

This paper advocates for the Stanford typed dependencies representation (henceforth SD) being a promising vehicle for bringing the breakthroughs of the last 15 years of parsing research to this broad potential user community. The representation aims to provide a simple, habitable design. All information is represented as binary relations. This maps straightforwardly on to common representations of potential users, including the logic forms of Moldovan and Rus (Moldovan and Rus, 2001),[2] semantic web Resource Description Framework (RDF) triples (http://www.w3.org/RDF/), and graph representations (with labeled edges and nodes). Unlike many linguistic formalisms, excessive detail is viewed as a defect: information that users do not understand or wish to process detracts from uptake and usability. The user-centered design process saw the key goal as representing semantically contentful relations suitable for relation extraction and more general information extraction uses. The design supports this use by favoring relations between content words, by maintaining semantically useful closed class word information while ignoring linguistic decisions less relevant to users, and by not representing less used material about linguistic features such as tense and agreement. The SD scheme thus provides a semantic representation simple and natural enough for people who are not (computational) linguists but can benefit from NLP tools.

2 Design choices and their implications

2.1 Design principles

The style of the SD representation bears a strong intellectual debt to the framework of Lexical-Functional Grammar (Bresnan, 2001), and, more directly, it owes a debt to both the sets of grammatical relations and the naming defined in two representations that follow an LFG style: the GR (Carroll et al., 1999) and PARC (King et al., 2003) schemes. These were used as a starting point for developing the Stanford dependencies (de Marneffe et al., 2006). But where the SD scheme deviates from GR, PARC, and its LFG roots is that it has been designed to be a practical model of sentence representation, particularly in the context of relation extraction tasks.

SD makes available two options, suited to different use cases: in one, every word of the original sentence is present as a node with relations between it and other nodes, whereas in the latter, certain words are "collapsed" out of the representation, making such changes as turning prepositions into relations. The former is useful when a close parallelism to the source text words must be maintained, whereas the latter is intended to be more useful for relation extraction and shallow language understanding tasks. Here, we discuss only the latter representation; see (de Marneffe et al., 2006) for a discussion of both options and the precise relationship between them.

The intended use cases of usability by people who are not (computational) linguists and suitability for relation extraction applications led SD to try to adhere to the following design principles (DPs):

1. Everything is represented uniformly as some binary relation between two sentence words.
2. Relations should be semantically contentful and useful to applications.
3. Where possible, relations should use notions of traditional grammar for easier comprehension by users.
4. Underspecified relations should be available to deal with the complexities of real text.
5. Where possible, relations should be between content words, not indirectly mediated via function words.
6. The representation should be spartan rather than overwhelming with linguistic details.

We illustrate many of them in the rest of this section, using example sentences which were made available for the Parser Evaluation Shared Task.

The grammatical relations of SD are arranged in a hierarchy, rooted with the most generic relation, *dependent*. The hierarchy contains 56 grammatical relations. When the relation between a head and its dependent can be identified more precisely, relations further down in the hierarchy are used, but when it is unclear, more generic dependencies are possible (DP1, DP4). For example, the *dependent* relation can be specialized to *aux* (auxiliary), *arg* (argument), or *mod* (modifier). The *arg* relation is further divided into the *subj* (subject) relation and the *comp* (complement) relation, and so on. The backbone of this hierarchy is quite similar to that in GR, but there are some crucial differences.

parser.

[2] The logic forms of Moldovan and Rus are in the form of a predicate calculus representation, although not one that represents such things as operator scope in a way that most would expect of a predicate calculus representation.

2.2 Comparison with GR and PARC

The SD scheme is not concerned with the argument/adjunct distinction which is largely useless in practice. In contrast, NP-internal relations are an inherent part of corpus texts and are critical in real-world applications. The SD scheme therefore includes many relations of this kind: *appos* (appositive modifier), *nn* (noun compound), *num* (numeric modifier), *number* (element of compound number) and *abbrev* (abbreviation), etc. (DP2). For instance, in the sentence *"I feel like a little kid," says a gleeful Alex de Castro, a car salesman, who has stopped by a workout of the Suns to slip six Campaneris cards to the Great Man Himself to be autographed* (*WSJ-R*), we obtain the following relations under the SD representation:

SD appos(Castro, salesman)
num(cards, six)
nn(cards, Campaneris)

The numeric modifier relation between *cards* and *six* is also standard in the PARC and GR schemes. PARC provides an apposition relation between *salesman* and *Alex de Castro*, whereas GR only identifies *salesman* as a text adjunct of *Castro*. But on the whole, SD makes more fine-grained distinctions in the relations, which are needed in practice. The *adjunct* dependency of the PARC scheme lumps together different relations. For example, the adjectival modifier *gleeful* in the sentence above will not be marked distinctively from the preposition modifying *workout*, nor from the relation between the verbs *stop* and *slip*:

PARC adjunct(Alex de Castro, gleeful)
adjunct(kid, little)
adjunct(stop, slip)
adjunct(workout, of)

The SD output for the relations between these words looks as follows:

SD amod(Castro, gleeful)
amod(kid, little)
xcomp(stop, slip)
prep_of(workout, Suns)

The comparison between the two outputs shows that SD proposes a larger set of dependencies, capturing relation differences which can play a role in applications (DP2), while sticking to notions of traditional grammar (DP3).

The SD scheme also chooses content words as heads of the dependencies (DP5). Auxiliaries, complementizers, and so on, are dependents of them. This choice in design is driven by the kind of information that is useful for applications. For instance, in the sentence *Considered as a whole, Mr. Lane said, the filings required under the proposed rules "will be at least as effective, if not more so, for investors following transactions" (WSJ-R)*, *effective* is chosen as the head of the quoted phrase. This enables the representation to have a direct dependency (*nsubj* for nominal subject) between the key content words *effective* and *filings*. Such a link is more difficult to infer from the GR scheme, where *be* is chosen as the head. However the relation between *effective* and *filings* is key to extracting the gist of the sentence semantics, and it is therefore important for applications to be able to retrieve it easily. Also, in the case of structures involving copular verbs, a direct link between the subject and the complement enables equivalent representations across languages (in Chinese, for example, copulas are not explicitly expressed). Such parallel representations should presumably help machine translation, and this was a further motivation for choosing content words as heads.

Another instance where direct links between content words is useful is the case of prepositional complements. The SD scheme offers the option of "collapsing" dependencies involving a preposition (DP5). In the example above, instead of having two relations *adjunct*(workout, of) and *obj*(of, Suns) as in PARC or *ncmod*(workout, of) and *dobj*(of, Suns) as in GR, SD provides a direct relation between the content words: *prep_of*(workout, Suns). Prepositions often work as role markers, and this type of link facilitates the extraction of how the two content words are related; and thus these links are often used by downstream applications (Lin and Pantel, 2001; Snow et al., 2005). The usefulness of the representation is exemplified in the sentence *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice (WSJ-R)* for which SD gives direct links between the entities joined through the preposition *such as*:

SD prep_such_as(crops, cotton)
prep_such_as(crops, soybeans)
prep_such_as(crops, rice)

A similar collapsing treatment takes place for conjuncts (DP5). Consider the following sentence: *Bell, based in Los Angeles, makes and distributes*

```
SD   nsubj(makes-8, Bell-1)
     nsubj(distributes-10, Bell-1)
     partmod(Bell-1, based-3)
     nn(Angeles-6, Los-5)
     prep_in(based-3, Angeles-6)
     conj_and(makes-8, distributes-10)
     amod(products-16, electronic-11)
     conj_and(electronic-11, computer-13)
     amod(products-16, computer-13)
     conj_and(electronic-11, building-15)
     amod(products-16, building-15)
     dobj(makes-8, products-16)
```

Figure 1: SD representation for *Bell, based in Los Angeles, makes and distributes electronic, computer and building products.*

```
GR   (passive based)
     (ncsubj based Bell obj)
     (ta bal Bell based)
     (iobj _ based in)
     (dobj in Angeles)
     (ncmod _ Angeles Los)
     (conj and makes)
     (conj and distributes)
     (conj and electronic)
     (conj and computer)
     (conj and building)
     (ncsubj and Bell _)
     (dobj and products)
     (ncmod _ products and)
```

Figure 2: GR representation for *Bell, based in Los Angeles, makes and distributes electronic, computer and building products.*

electronic, computer and building products (WSJ-R). Figures 1 and 2 give the full dependency output from SD and GR, respectively. The numbers after the words in the SD representation indicate the word position in the sentence.[3] From the SD representation, one can easily see that the sentence talks about *electronic products* and *computer products* as well as *building products*. By collapsing the dependencies involving conjuncts, the output produced is closer to the semantics of the sentence, and this facilitates information extraction (DP2). This information is not straightforwardly apparent in the GR scheme (see figure 2), nor in the PARC scheme which follows a similar treatment of conjuncts.

Another choice in the design has been to consistently have binary relations (DP1). All the dependencies form a triple: a grammatical relation holding between two words (head and dependent). This gives uniformity to the representation and renders it very readable, critical features for a user-centered design. Furthermore, all the information can be represented by a directed graph, enabling the creation of both a limpid visual representation for humans and a canonical data structure for software. Moreover, it maps straightforwardly on to semantic web representations such as OWL and RDF triples, as exploited in (Zouaq et al., 2006; Zouaq et al., 2007).

This design choice limits the kind of information offered by the SD scheme. For instance, the PARC scheme contains much more information about individual words, such as verb tense and aspect, noun number and person, type of NE for proper nouns, pronoun form, adjective degree, etc. For the sentence in figures 1 and 2, the following information is available for the word *Los Angeles* in the PARC scheme:

```
PARC   num(Los Angeles∼5, sg)
       pers(Los Angeles∼5, 3)
       proper(Los Angeles∼5, location)
```

This kind of information is indubitably valuable, but is often less used in practice, and does not per se pertain to dependency data. Adding it lengthens an output already complex enough, and impedes readability and convenience. Thus, SD does not provide such overwhelming detail (DP6).

2.3 Trading off linguistic fidelity and usability

We feel that turning prepositions into relations is useful for 98% of users 98% of the time. Nevertheless opting for usability in this way causes the SD scheme to sacrifice some linguistic fidelity. One instance is that modifiers of prepositions are dependent on the verb (or more precisely, on the head of the clause in which they appear) and not on the preposition itself. In *Bill went over the river and right through the woods*, *right* will be an adverbial modifier of *went*. In *He had laughed, simultaneously mocking the stupidity of government by cosmetics and confessing that he was also a part of it, just as he was part of government by voice coach and acting coach (BNC)*, *just* which modifies *as* will be a dependent of the head of the adverbial

[3] Without word position, the representation is deficient if the same word occurs more than once in a sentence.

clause, i.e., *part*. This induces some distortion in the exact semantics of the sentence.

The interaction between preposition collapsing and PP conjunction is another instance in which the SD treatment slightly alters the semantics of the sentence. Consider again the sentence *Bill went over the river and right through the woods*. Both prepositions, *over* and *through*, are governed by the verb *went*. To avoid disjoint subgraphs when collapsing the relations, examples like this are transformed into VP coordination, which requires making a copy of the word *went*. This gives the following representation, which corresponds to a sentence like *Bill went over the river and went right through the woods*:

SD prep_over(went-2, river-5)
prep_through(went-2', woods-10)
conj_and(went-2, went-2')

Not collapsing the relations in such a case would prevent the alteration of the semantics, but would lead to a non-uniform treatment of prepositions. Uniformity is key for readability and user convenience. It seems therefore reasonable to use a representation which sacrifices the exact semantics of the original sentence by producing a sentence roughly equivalent, but which ensures uniformity across relations.

3 The formalism and the tool

Two vital conditions for the success of a dependency scheme are to provide a suitable representation for users as well as a tool that is easy to use. Sagae et al. (2008) note that the availability of an automatic procedure to convert phrase structure parses to SD is the reason for its use in evaluations of parsers in the biomedical domain. The primary focus of the SD scheme, however, has been to offer grammatical relations appropriate for end-users.

The Stanford parser[4] comes with a tool, described in (de Marneffe et al., 2006), which provides for the rapid extraction of the grammatical relations from phrase structure parses. Structural configurations are used to define grammatical roles: the semantic head of each constituent of the parse is identified, using rules akin to the Collins head rules, but modified to retrieve the semantic head of the constituent rather than the syntactic head. As mentioned, content words are chosen as heads, and all the other words in the constituent depend on this head. To retrieve adequate heads from a semantic point of view, heuristics are used to inject more structure when the Penn Treebank gives only flat constituents, as is often the case for conjuncts, e.g., (NP the new phone book and tour guide), and QP constituents, e.g., (QP more than 300). Then for each grammatical relation, patterns are defined over the phrase structure parse tree using the tree-expression syntax defined by tregex (Levy and Andrew, 2006). Conceptually, each pattern is matched against every tree node, and the matching pattern with the most specific grammatical relation is taken as the type of the dependency.

The automatic extraction of the relations is not infallible. For instance, in the sentence *Behind their perimeter walls lie freshly laundered flowers, verdant grass still sparkling from the last shower, yew hedges in an ecstasy of precision clipping (BNC)*, the system will erroneously retrieve apposition relations between *flowers* and *grass*, as well as between *flowers* and *hedges* whereas these should be *conj_and* relations. The system is clueless when there is no overt maker of conjunction.

Another limitation of the tool is the treatment of long-distance dependencies, such as *wh*-movement and control/raising: the system cannot handle long-distance dependencies that cross clauses. In a sentence like *What does he think?*, the system will correctly find that *what* is a direct object of *think*:

SD dobj(think-4, What-1)
aux(think-4, does-2)
nsubj(think-4, he-3)

However in a sentence such as *Who the hell does he think he's kidding? (BNC)*, the automatic extraction will fail to find that *who* is the direct object of *kidding*. Here, it is vital to distinguish between SD as a representation versus the extant conversion tool. Long-distance dependencies are not absent from the formalism, but the tool does not accurately deal with them.[5]

4 Stanford dependencies in practice

SD has been successfully used by researchers in different domains. In the PASCAL Recognizing

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] As possible future work, we have thought of using a tool such as the one of Levy and Manning (2004) to correctly determine long distance dependencies, as input to the current dependency conversion system. This would presumably be effective, but would make the conversion process much heavier weight.

Textual Entailment (RTE) challenges (Dagan et al., 2006; Giampiccolo et al., 2007), the increase in the use of SD is clearly apparent. The goal in these challenges consists of identifying whether one sentence follows from a piece of text and general background knowledge, according to the intuitions of an intelligent human reader. In 2007, out of the 21 systems which participated in the challenge, 5 used the SD representation, whereas the year before only the Stanford entry was using it.

SD is also widely present in the bioinformatic world where it is used with success (Erkan et al., 2007; Greenwood and Stevenson, 2007; Urbain et al., 2007; Clegg, 2008). Fundel et al. (2007) found that, in extraction of relations between genes and proteins, a system based on the SD scheme greatly outperformed the previous best system on the LLL challenge dataset (by an 18% absolute improvement in F-measure). Airola et al. (2008) provide more systematic results on a number of protein-protein interaction datasets. Their graph kernel approach uses an all-dependency-paths kernel which allows their system to consider full dependency graphs. Their system is based on the SD scheme, and they demonstrate state-of-the-art performance for this approach.

In the biomedical domain, SD has recently been used in evaluations of parsers (Clegg and Shepherd, 2007; Pyysalo et al., 2007a). Pyysalo et al. (2007a) assessed the suitability of the SD scheme over the Link Grammar dependency scheme in an application-oriented evaluation. The Link Parser indeed uses a very fine-grained set of relations, which often makes distinctions of a structural rather than a semantic nature. One example is the MX relation which "connects modifying phrases with commas to preceding nouns ('The DOG, a POODLE, was black'; 'JOHN, IN a black suit, looked great')." The Link Parser uses a different set of dependency types for dependencies appearing in questions and relative clauses. Another example is the prepositional phrase where alternative attachment structures are indicated by different relations. Many of these distinctions are too fine and non-semantic to be of practical value. The SD scheme, by aiming for an intermediate level of granularity, and targeting semantic dependencies, provides a more adequate representation for applications. Therefore, to increase the usability of the BioInfer corpus (Pyysalo et al., 2007b), which provides manually annotated data for information extraction in the biomedical domain and originally followed the Link Grammar scheme, Pyysalo et al. (2007a) developed a version of the corpus annotated with the SD scheme. They also made available a program and conversion rules that they used to transform Link Grammar relations into SD graphs, which were then hand-corrected (Pyysalo et al., 2007b). While a limited amount of gold standard annotated data was prepared for the Parser Evaluation Shared Task, this is the main source of gold-standard SD data which is currently available.

In other domains, Zhuang et al. (2006) uses the representation to extract opinions about features in reviews and Meena and Prabhakar (2007) uses it to improve the quality of sentence-level sentiment analysis. The open information extraction system TEXTRUNNER (Banko et al., 2007) also makes use of the SD graph representation: its first module uses the Stanford parser and the dependency tool to automatically identify and label trustworthy and untrustworthy extractions. Even in theoretical linguistic work, SD has proven very useful: it has hugely facilitated data extraction from corpora, in the context of the NSF-funded project "Dynamics of probabilistic grammar" carried out at the Stanford Linguistics department.

5 Suitability for parser evaluation

When seeking a gold-standard dependency scheme for parser evaluation, the ultimate goal of such an evaluation is an important question. It is necessary to contrast the two different forms that evaluation can take: extrinsic task-based evaluation and intrinsic evaluation. We tend to agree with Mollá and Hutchinson (2003) that intrinsic evaluations have limited value and that task-based evaluation is the correct approach. Some of the results of the previous section at least broadly support the utility of the SD scheme for practical use in higher-level tasks. Nevertheless, given the current trend in the NLP community as well as in other fields such as bioinformatics, where the advantage of dependency representations for shallow text understanding tasks has become salient, we would argue, following Clegg and Shepherd (2007), that dependency-based evaluation is close to typical user tasks. Moreover, it avoids some of the known deficiencies of other parser evaluation measures such as Parseval (Carroll et al., 1999).

Recent work on parser evaluation using dependency graphs in the biomedical domain confirms that researchers regard dependency-based evaluation as a more useful surrogate for extrinsic task-based evaluation (Clegg and Shepherd, 2007; Pyysalo et al., 2007a). In their evaluation, Clegg and Shepherd (2007) aimed at analyzing the capabilities of syntactic parsers with respect to semantically important tasks crucial to biological information extraction systems. To do so, they used the SD scheme, which provides "a de facto standard for comparing a variety of constituent parsers and treebanks at the dependency level," and they assessed its suitability for evaluation. They found that the SD scheme better illuminates the performance differences between higher ranked parsers (e.g., Charniak-Lease parser (Lease and Charniak, 2005)), and lower ranked parsers (e.g., the Stanford parser (Klein and Manning, 2003)). Their parser evaluation accommodates user needs: they used the collapsed version of the dependency graphs offered by the SD scheme, arguing that this is the kind of graph one would find most useful in an information extraction project. Although Clegg and Shepherd (2007) also favor dependency graph representations for parser evaluation, they advocate retention of parse trees so information lost in the dependency structures can be accessed.

In essence, any existing dependency scheme could be adopted as the gold-standard for evaluation. However if one believes in ultimately valuing extrinsic task-based evaluation, a dependency representation which proposes a suitable design for users and user tasks is probably the best surrogate for intrinsic evaluation. Moreover, the existence of tools for automatically generating and converting dependency representations has aided greatly in making parser comparison possible across different formalisms. We believe that the SD scheme approaches these goals. If one accepts the goals set here, in order to enforce uniformity between application and evaluation, it seems sensible to have a unique scheme for both purposes. Some of the positive results from use of the SD representation, as well as the evaluations carried out in the biomedical field, point to the usability of the SD scheme for both purposes.

Acknowledgments

We wish to thank Andrew Brian Clegg and Sampo Pyysalo for their useful feedback on the dependency extraction tool. Their comments enabled us to improve the tool. We also thank the workshop reviewers for their helpful comments.

References

Airola, Antti, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. A graph kernel for protein-protein interaction extraction. In *Proceedings of BioNLP 2008: Current Trends in Biomedical Natural Language Processing (ACL08)*.

Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*.

Black, E., S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings, Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA. DARPA.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Blackwell, Oxford.

Carroll, John, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC)*.

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

Clegg, Andrew B. 2008. *Computational-Linguistic Approaches to Biological Text Mining*. Ph.D. thesis, School of Crystallography, Birkbeck, University of London.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In et al., Quinonero-Candela, editor, *MLCW 2005, LNAI Volume 3944*, pages 177–190. Springer-Verlag.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*.

Erkan, Gunes, Arzucan Ozgur, and Dragomir R. Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Fellbaum, Christiane. 1998. *WordNet: an electronic lexical database*. MIT Press.

Fillmore, Charles J., Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

Fundel, Katrin, Robert Küffner, and Ralf Zimmer. 2007. RelEx relation extraction using dependency parse trees. *Bioinformatics*, 23.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Greenwood, Mark A. and Mark Stevenson. 2007. A semi-supervised approach to learning relevant protein-protein interaction articles. In *Proceedings of the Second BioCreAtIvE Challenge Workshop, Madrid, Spain*.

King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 dependency bank. In *4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Lease, Matthew and Eugene Charniak. 2005. Parsing biomedical literature. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP'05)*.

Levy, Roger and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *LREC 2006*. http://www-nlp.stanford.edu/software/tregex.shtml.

Levy, Roger and Christopher D. Manning. 2004. Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation. In *ACL 42*, pages 328–335.

Lin, Dekang and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Lin, Dekang. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems, Granada, Spain*.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19 (2).

Meena, Arun and T. V. Prabhakar. 2007. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *Advances in Information Retrieval*, volume 4425 of *Lecture Notes in Computer Science*. Springer.

Moldovan, Dan I. and Vasile Rus. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Meeting of the Association for Computational Linguistics*, pages 394–401.

Mollá, Diego and Ben Hutchinson. 2003. Intrinsic versus extrinsic evaluations of parsing systems. In *Proceedings of the Workshop on Evaluation Initiatives in Natural Language Processing*, pages 43–50. European Association for Computational Linguistics.

Pyysalo, Sampo, Filip Ginter, Katri Haverinen, Juho Heimonen, Tapio Salakoski, and Veronika Laippala. 2007a. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proceedings of BioNLP 2007: Biological, translational, and clinical language processing (ACL07)*.

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.

Sagae, Kenji, Yusuke Miyao, and Jun'ichi Tsujii. 2008. Challenges in mapping of syntactic representations for framework-independent parser evaluation. In *Proceedings of the Workshop on Automated Syntatic Annotations for Interoperable Language Resources at the First International Conference on Global Interoperability for Language Resources (ICGL'08)*.

Sleator, Daniel D. and Davy Temperley. 1993. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*.

Snow, Rion, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of NIPS 2004*.

Urbain, Jay, Nazli Goharian, and Ophir Frieder. 2007. IIT TREC 2007 genomics track: Using concept-based semantics in context for genomics literature passage retrieval. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*.

Zhuang, Li, Feng Jing, Xiao yan Zhu, and Lei Zhang. 2006. Movie review mining and summarization. In *Proc. ACM Conference on Information and Knowledge Management (CIKM)*.

Zouaq, Amal, Roger Nkambou, and Claude Frasson. 2006. The knowledge puzzle: An integrated approach of intelligent tutoring systems and knowledge management. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006)*, pages 575–582.

Zouaq, Amal, Roger Nkambou, and Claude Frasson. 2007. Building domain ontologies from text for educational purposes. In *Proceedings of the Second European Conference on Technology Enhanced Learning: Creating new learning experiences on a global scale*.

Exploring an Auxiliary Distribution based approach to Domain Adaptation of a Syntactic Disambiguation Model

Barbara Plank
University of Groningen
The Netherlands
B.Plank@rug.nl

Gertjan van Noord
University of Groningen
The Netherlands
G.J.M.van.Noord@rug.nl

Abstract

We investigate auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation of a supervised parsing system of Dutch. To overcome the limited target domain training data, we exploit an original and larger out-of-domain model as auxiliary distribution. However, our empirical results exhibit that the auxiliary distribution does not help: even when very little target training data is available the incorporation of the out-of-domain model does not contribute to parsing accuracy on the target domain; instead, better results are achieved either without adaptation or by simple model combination.

1 Introduction

Modern statistical parsers are trained on large annotated corpora (treebanks) and their parameters are estimated to reflect properties of the training data. Therefore, a disambiguation component will be successful as long as the treebank it was trained on is representative for the input the model gets. However, as soon as the model is applied to another *domain*, or *text genre* (Lease et al., 2006), accuracy degrades considerably. For example, the performance of a parser trained on the Wall Street Journal (newspaper text) significantly drops when evaluated on the more varied Brown (fiction/nonfiction) corpus (Gildea, 2001).

A simple solution to improve performance on a new domain is to construct a parser specifically for that domain. However, this amounts to hand-labeling a considerable amount of training data which is clearly very expensive and leads to an unsatisfactory solution. In alternative, techniques for *domain adaptation*, also known as *parser adaptation* (McClosky et al., 2006) or *genre portability* (Lease et al., 2006), try to leverage either a small amount of already existing annotated data (Hara et al., 2005) or unlabeled data (McClosky et al., 2006) of one domain to parse data from a different domain. In this study we examine an approach that assumes a limited amount of already annotated in-domain data.

We explore auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation, originally suggested for the incorporation of lexical selectional preferences into a parsing system. We gauge the effect of exploiting a more general, out-of-domain model for parser adaptation to overcome the limited amount of in-domain training data. The approach is examined on two application domains, question answering and spoken data.

For the empirical trials, we use Alpino (van Noord and Malouf, 2005; van Noord, 2006), a robust computational analyzer for Dutch. Alpino employs a discriminative approach to parse selection that bases its decision on a Maximum Entropy (MaxEnt) model. Section 2 introduces the MaxEnt framework. Section 3 describes our approach of exploring auxiliary distributions for domain adaptation. In section 4 the experimental design and empirical results are presented and discussed.

2 Background: MaxEnt Models

Maximum Entropy (MaxEnt) models are widely used in Natural Language Processing (Berger et al., 1996; Ratnaparkhi, 1997; Abney, 1997). In this framework, a disambiguation model is speci-

Proceedings of the Coling 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 9–16
Manchester, August 2008

fied by a set of feature functions describing properties of the data, together with their associated weights. The weights are learned during the training procedure so that their estimated value determines the contribution of each feature. In the task of parsing, features appearing in correct parses are given increasing weight, while features in incorrect parses are given decreasing weight. Once a model is trained, it can be applied to parse selection that chooses the parse with the highest sum of feature weights.

During the training procedure, the weights vector is estimated to best fit the training data. In more detail, given m features with their corresponding empirical expectation $E_{\tilde{p}}[f_j]$ and a default model q_0, we seek a model p that has minimum Kullback-Leibler (KL) divergence from the default model q_0, subject to the expected-value constraints: $E_p[f_j] = E_{\tilde{p}}[f_j]$, where $j \in 1, ..., m$.

In MaxEnt estimation, the default model q_0 is often only implicit (Velldal and Oepen, 2005) and not stated in the model equation, since the model is assumed to be uniform (e.g. the constant function $\frac{1}{\Omega(s)}$ for sentence s, where $\Omega(s)$ is the set of parse trees associated with s). Thus, we seek the model with minimum KL divergence from the uniform distribution, which means we search model p with maximum entropy (uncertainty) subject to given constraints (Abney, 1997).

In alternative, if q_0 is not uniform then p is called a *minimum divergence model* (according to (Berger and Printz, 1998)). In the statistical parsing literature, the default model q_0 that can be used to incorporate prior knowledge is also referred to as base model (Berger and Printz, 1998), default or reference distribution (Hara et al., 2005; Johnson et al., 1999; Velldal and Oepen, 2005).

The solution to the estimation problem of finding distribution p, that satisfies the expected-value constraints and minimally diverges from q_0, has been shown to take a specific parametric form (Berger and Printz, 1998):

$$p_\theta(\omega, s) = \frac{1}{Z_\theta} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{1}$$

with m feature functions, s being the input sentence, ω a corresponding parse tree, and Z_θ the normalization equation:

$$Z_\theta = \sum_{\omega' \in \Omega} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega')} \tag{2}$$

Since the sum in equation 2 ranges over all possible parse trees $\omega' \in \Omega$ admitted by the grammar, calculating the normalization constant renders the estimation process expensive or even intractable (Johnson et al., 1999). To tackle this problem, Johnson et al. (1999) redefine the estimation procedure by considering the conditional rather than the joint probability.

$$P_\theta(\omega|s) = \frac{1}{Z_\theta} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{3}$$

with Z_θ as in equation 2, but instead, summing over $\omega' \in \Omega(s)$, where $\Omega(s)$ is the set of parse trees associated with sentence s. Thus, the probability of a parse tree is estimated by summing only over the possible parses of a specific sentence.

Still, calculating $\Omega(s)$ is computationally very expensive (Osborne, 2000), because the number of parses is in the worst case exponential with respect to sentence length. Therefore, Osborne (2000) proposes a solution based on *informative samples*. He shows that is suffices to train on an informative subset of available training data to accurately estimate the model parameters. Alpino implements the Osborne-style approach to Maximum Entropy parsing. The standard version of the Alpino parser is trained on the Alpino newspaper Treebank (van Noord, 2006).

3 Exploring auxiliary distributions for domain adaptation

3.1 Auxiliary distributions

Auxiliary distributions (Johnson and Riezler, 2000) offer the possibility to incorporate information from additional sources into a MaxEnt Model. In more detail, auxiliary distributions are integrated by considering the logarithm of the probability given by an auxiliary distribution as an additional, real-valued feature. More formally, given k auxiliary distributions $Q_i(\omega)$, then k new *auxiliary features* $f_{m+1}, ..., f_{m+k}$ are added such that

$$f_{m+i}(\omega) = \log Q_i(\omega) \tag{4}$$

where $Q_i(\omega)$ do not need to be proper probability distributions, however they must strictly be positive $\forall \omega \in \Omega$ (Johnson and Riezler, 2000).

The auxiliary distributions resemble a reference distribution, but instead of considering a single reference distribution they have the advantage that several auxiliary distributions can be integrated and weighted against each other. Johnson establishes the following equivalence between the two (Johnson and Riezler, 2000; Velldal and Oepen, 2005):

$$Q(\omega) = \prod_{i=1}^{k} Q_i(\omega)^{\theta_{m+i}} \tag{5}$$

where $Q(\omega)$ is the reference distribution and $Q_i(\omega)$ is an auxiliary distribution. Hence, the contribution of each auxiliary distribution is regulated through the estimated feature weight. In general, a model that includes k auxiliary features as given in equation (4) takes the following form (Johnson and Riezler, 2000):

$$P_\theta(\omega|s) = \frac{\prod_{i=1}^{k} Q_i(\omega)^{\theta_{m+i}}}{Z_\theta} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{6}$$

Due to the equivalence relation in equation (5) we can restate the equation to explicitly show that auxiliary distributions are additional features[1].

$$P_\theta(\omega|s)$$

$$= \frac{\prod_{i=1}^{k} [exp^{f_{m+i(\omega)}}]^{\theta_{m+i}}}{Z_\theta} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{7}$$

$$= \frac{1}{Z_\theta} \prod_{i=1}^{k} exp^{f_{m+i(\omega)} * \theta_{m+i}} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{8}$$

$$= \frac{1}{Z_\theta} exp^{\sum_{i=1}^{k} f_{m+i(\omega)} * \theta_{m+i}} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \tag{9}$$

$$= \frac{1}{Z_\theta} exp^{\sum_{j=1}^{m+k} \theta_j f_j(\omega)} \quad \text{with } f_j(\omega) = logQ(\omega) \text{ for } m < j \leq (m+k) \tag{10}$$

3.2 Auxiliary distributions for adaptation

While (Johnson and Riezler, 2000; van Noord, 2007) focus on incorporating several auxiliary distributions for lexical selectional preferences, in this study we explore auxiliary distributions for domain adaptation.

We exploit the information of the more general model, estimated from a larger, out-of-domain treebank, for parsing data from a particular target domain, where only a small amount of training data is available. A related study is Hara et al. (2005). While they also assume a limited amount of in-domain training data, their approach differs from ours in that they incorporate an original model as a reference distribution, and their estimation procedure is based on parse forests (Hara et al., 2005; van Noord, 2006), rather than informative samples. In this study, we want to gauge the effect of auxiliary distributions, which have the advantage that the contribution of the additional source is regulated.

More specifically, we extend the target model to include (besides the original integer-valued features) one additional real-valued feature (k=1)[2]. Its value is defined to be the negative logarithm of the conditional probability given by OUT, the original, out-of-domain, Alpino model. Hence, the general model is 'merged' into a single auxiliary feature:

$$f_{m+1} = -logP_{OUT}(\omega|s) \tag{11}$$

The parameter of the new feature is estimated using the same estimation procedure as for the remaining model parameters. Intuitively, our auxiliary feature models dispreferences of the general model for certain parse trees. When the Alpino model assigns a high probability to a parse candidate, the auxiliary feature value will be small, close to zero. In contrast, a low probability parse tree in the general model gets a higher feature value. Together with the estimated feature weight expected to be negative, this has the effect that a low probability parse in the Alpino model will reduce the probability of a parse in the target domain.

3.3 Model combination

In this section we sketch an alternative approach where we keep only two features under the MaxEnt framework: one is the log probability assigned by the out-domain model, the other the log probability assigned by the in-domain model:

$$f_1 = -logP_{OUT}(\omega|s), f_2 = -logP_{IN}(\omega|s)$$

The contribution of each feature is again scaled through the estimated feature weights θ_1, θ_2.
We can see this as a simple instantiation of *model combination*. In alternative, *data combination* is a domain adaptation method where IN and OUT-domain data is simply concatenated and a new model trained on the union of data. A potential and well known disadvantage of data combination is that the usually larger amount of out-domain data

[1] Note that the step from equation (6) to (7) holds by restating equation (4) as $Q_i(\omega) = exp^{f_{m+i}(\omega)}$

[2] Or alternatively, $k \geq 1$ (see section 4.3.1).

’overwhelms’ the small amount of in-domain data. Instead, Model combination interpolates the two *models* in a linear fashion by scaling their contribution. Note that if we skip the parameter estimation step and simply assign the two parameters equal values (equal weights), the method reduces to $P_{OUT}(\omega|s) \times P_{IN}(\omega|s)$, i.e. just multiplying the respective model probabilities.

4 Experiments and Results

4.1 Experimental design

The general model is trained on the Alpino Treebank (van Noord, 2006) (newspaper text; approximately 7,000 sentences). For the domain-specific corpora, in the first set of experiments (section 4.3) we consider the Alpino CLEF Treebank (questions; approximately 1,800 sentences). In the second part (section 4.4) we evaluate the approach on the Spoken Dutch corpus (Oostdijk, 2000) (CGN, ’Corpus Gesproken Nederlands’; spoken data; size varies, ranging from 17 to 1,193 sentences). The CGN corpus contains a variety of components/subdomains to account for the various dimensions of language use (Oostdijk, 2000).

4.2 Evaluation metric

The output of the parser is evaluated by comparing the generated dependency structure for a corpus sentence to the gold standard dependency structure in a treebank. For this comparison, we represent the dependency structure (a directed acyclic graph) as a set of named dependency relations. To compare such sets of dependency relations, we count the number of dependencies that are identical in the generated parse and the stored structure, which is expressed traditionally using precision, recall and f-score (Briscoe et al., 2002).

Let D_p^i be the number of dependencies produced by the parser for sentence i, D_g^i is the number of dependencies in the treebank parse, and D_o^i is the number of correct dependencies produced by the parser. If no superscript is used, we aggregate over all sentences of the test set, i.e.,:

$$D_p = \sum_i D_p^i \qquad D_o = \sum_i D_o^i \qquad D_g = \sum_i D_g^i$$

Precision is the total number of correct dependencies returned by the parser, divided by the overall number of dependencies returned by the parser (precision $= D_o/D_p$); recall is the number of correct system dependencies divided by the total number of dependencies in the treebank (recall $= D_o/D_g$). As usual, precision and recall can be combined in a single f-score metric.

An alternative similarity score for dependency structures is based on the observation that for a given sentence of n words, a parser would be expected to return n dependencies. In such cases, we can simply use the percentage of correct dependencies as a measure of accuracy. Such a labeled dependency accuracy is used, for instance, in the CoNLL shared task on dependency parsing (“labeled attachment score”).

Our evaluation metric is a variant of labeled dependency accuracy, in which we do allow for some discrepancy between the number of returned dependencies. Such a discrepancy can occur, for instance, because in the syntactic annotations of Alpino (inherited from the CGN) words can sometimes be dependent on more than a single head (called ‘secondary edges’ in CGN). A further cause is parsing failure, in which case a parser might not produce any dependencies. We argue elsewhere (van Noord, In preparation) that a metric based on f-score can be misleading in such cases. The resulting metric is called *concept accuracy*, in, for instance, Boros et al. (1996).[3]

$$\text{CA} = \frac{D_o}{\sum_i \max(D_g^i, D_p^i)}$$

The concept accuracy metric can be characterized as the mean of a per-sentence minimum of recall and precision. The resulting CA score therefore is typically slightly lower than the corresponding f-score, and, for the purposes of this paper, equivalent to labeled dependency accuracy.

4.3 Experiments with the QA data

In the first set of experiments we focus on the Question Answering (QA) domain (CLEF corpus). Besides evaluating our auxiliary based approach (section 3), we conduct separate baseline experiments:

- **In-domain (CLEF):** train on CLEF (baseline)
- **Out-domain (Alpino):** train on Alpino
- **Data Combination (CLEF+Alpino):** train a model on the combination of data, CLEF $\cup$ Alpino

[3] In previous publications and implementations definitions were sometimes used that are equivalent to: CA $= \frac{D_o}{\max(D_g, D_p)}$ which is slightly different; in practice the differences can be ignored.

Dataset	In-dom.	Out-dom.	Data Combination	Aux.distribution	Model Combination	
size (#sents)	CLEF	Alpino	CLEF+Alpino	CLEF+Alpino_aux	CLEF_aux+Alpino_aux	equal weights
CLEF 2003 (446)	97.01	94.02	$\underline{97.21}$	97.01	$\underline{97.14}$	$\underline{97.46}$
CLEF 2004 (700)	96.60	89.88	95.14	96.60	$\underline{97.12}$	$\underline{97.23}$
CLEF 2005 (200)	97.65	87.98	93.62	$\underline{97.72}$	$\underline{97.99}$	$\underline{98.19}$
CLEF 2006 (200)	97.06	88.92	95.16	97.06	97.00	96.45
CLEF 2007 (200)	96.20	92.48	$\underline{97.30}$	$\underline{96.33}$	$\underline{96.33}$	$\underline{96.46}$

Table 1: Results on the CLEF test data; underlined scores indicate results $>$ in-domain baseline (CLEF)

- **Auxiliary distribution (CLEF+Alpino_aux):** adding the original Alpino model as auxiliary feature to CLEF

- **Model Combination:** keep only two features $P_{OUT}(\omega|s)$ and $P_{IN}(\omega|s)$. Two variants: i) estimate the parameters θ_1, θ_2 (**CLEF_aux+Alpino_aux**); ii) give them equal values, i.e. θ_1=θ_2=-1 (**equal weights**)

We assess the performance of all of these models on the CLEF data by using 5-fold cross-validation. The results are given in table 1.

The CLEF model performs significantly better than the out-of-domain (Alpino) model, despite of the smaller size of the in-domain training data. In contrast, the simple data combination results in a model (CLEF+Alpino) whose performance is somewhere in between. It is able to contribute in some cases to disambiguate questions, while leading to wrong decisions in other cases.

However, for our auxiliary based approach (CLEF+Alpino_aux) with its regulated contribution of the general model, the results show that adding the feature does not help. On most datasets the same performance was achieved as by the in-domain model, while on only two datasets (CLEF 2005, 2007) the use of the auxiliary feature results in an insignificant improvement.

In contrast, simple model combination works surprisingly well. On two datasets (CLEF 2004 and 2005) this simple technique reaches a substantial improvement over *all* other models. On only one dataset (CLEF 2006) it falls slightly off the in-domain baseline, but still considerably outperforms data combination. This is true for both model combination methods, with estimated and equal weights. In general, the results show that model combination usually outperforms data combination (with the exception of one dataset, CLEF 2007), where, interestingly, the simplest model combination (equal weights) often performs best.

Contrary to expectations, the auxiliary based approach performs poorly and could often not even come close to the results obtained by simple model combination. In the following we will explore possible reasons for this result.

Examining possible causes One possible point of failure could be that the auxiliary feature was simply ignored. If the estimated weight would be close to zero the feature would indeed not contribute to the disambiguation task. Therefore, we examined the estimated weights for that feature. From that analysis we saw that, compared to the other features, the auxiliary feature got a weight relatively far from zero. It got on average a weight of -0.0905 in our datasets and as such is among the most influential weights, suggesting it to be important for disambiguation.

Another question that needs to be asked, however, is whether the feature is modeling properly the original Alpino model. For this sanity check, we create a model that contains only the single auxiliary feature and no other features. The feature's weight is set to a constant negative value[4]. The resulting model's performance is assessed on the complete CLEF data. The results (0% column in table 3) show that the auxiliary feature is indeed properly modeling the general Alpino model, as the two result in identical performance.

4.3.1 Feature template class models

In the experiments so far the general model was 'packed' into a single feature value. To check whether the feature alone is too weak, we examine the inclusion of several auxiliary distributions ($k > 1$). Each auxiliary feature we add represents a 'submodel' corresponding to an actual feature template class used in the original model. The feature's value is the negative log-probability as defined in equation 11, where OUT corresponds to the respective Alpino submodel.

The current Disambiguation Model of Alpino uses the 21 feature templates (van Noord and Malouf, 2005). Out of this given feature templates, we create two models that vary in the number of classes used. In the first model ('5 class'), we create five ($k = 5$) auxiliary distributions corresponding to five clusters of feature templates. They are

[4] Alternatively, we may estimate its weight, but as it does not have competing features we are safe to assume it constant.

defined manually and correspond to submodels for Part-of-Speech, dependencies, grammar rule applications, bilexical preferences and the remaining Alpino features. In the second model ('21 class'), we simply take every single feature template as its own cluster ($k = 21$).

We test the two models and compare them to our baseline. The results of this experiment are given in table 2. We see that both the 5 class and the 21 class model do not achieve any considerable improvement over the baseline (CLEF), nor over the single auxiliary model (CLEF+Alpino_aux).

Dataset (#sents)	5class	21class	CLEF+Alpino_aux	CLEF
CLEF2003 (446)	97.01	97.04	97.01	97.01
CLEF2004 (700)	96.57	96.60	96.60	96.60
CLEF2005 (200)	97.72	97.72	97.72	97.65
CLEF2006 (200)	97.06	97.06	97.06	97.06
CLEF2007 (200)	96.20	96.27	96.33	96.20

Table 2: Results on CLEF including several auxiliary features corresponding to Alpino submodels

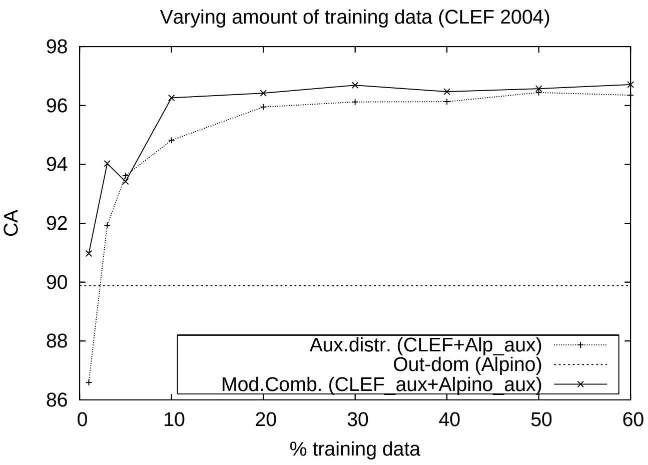


Figure 1: Amount of in-domain training data versus concept accuracy (Similar figures result from the other CLEF datasets) - note that we depict only aux.distr. as its performance is nearly indistinguishable from the in-domain (CLEF) baseline

4.3.2 Varying amount of training data

Our expectation is that the auxiliary feature is at least helpful in the case very little in-domain training data is available. Therefore, we evaluate the approach with smaller amounts of training data.

We sample (without replacement) a specific amount of training instances from the original QA data files and train models on the reduced training data. The resulting models are tested with and without the additional feature as well as model combination on the complete data set by using cross validation. Table 3 reports the results of these experiments for models trained on a proportion of up to 10% CLEF data. Figure 1 illustrates the overall change in performance.

Obviously, an increasing amount of in-domain training data improves the accuracy of the models. However, for our auxiliary feature, the results in table 3 show that the models with and without the auxiliary feature result in an overall almost identical performance (thus in figure 1 we depict only one of the lines). Hence, the inclusion of the auxiliary feature does not help in this case either. The models achieve similar performance even independently of the available amount of in-domain training data.

Thus, even on models trained on very little in-domain training data (e.g. 1% CLEF training data) the auxiliary based approach does not work. It even hurts performance, i.e. depending on the specific dataset, the inclusion of the auxiliary feature results in a model whose performance lies even *below* the original Alpino model accuracy, for up to a certain percentage of training data (varying on the dataset from 1% up to 10%).

In contrast, simple model combination is much more beneficial. It is able to outperform almost constantly the in-domain baseline (CLEF) and our auxiliary based approach (CLEF+Alpino_aux). Furthermore, in contrast to the auxiliary based approach, model combination never falls below the out-of-domain (Alpino) baseline, not even in the case a tiny amount of training data is available. This is true for both model combinations (estimated versus equal weights).

We would have expected the auxiliary feature to be useful at least when very little in-domain training data is available. However, the empirical results reveal the contrary[5]. We believe the reason for this drop in performance is the amount of available in-domain training data and the corresponding scaling of the auxiliary feature's weight. When little training data is available, the weight cannot be estimated reliably and hence is not contributing enough compared to the other features (exemplified in the drop of performance from 0% to 1%

[5] As suspected by a reviewer, the (non-auxiliary) features may overwhelm the single auxiliary feature, such that possible improvements by increasing the feature space on such a small scale might be invisible. We believe this is not the case. Other studies have shown that including just a few features might indeed help (Johnson and Riezler, 2000; van Noord, 2007). (e.g., the former just added 3 features).

	0%		1%				5%				10%			
Dataset	no aux	= Alp.	no aux	+aux	m.c.	eq.w.	no aux	+aux	m.c.	eq.w.	no aux	+aux	m.c.	eq.w.
CLEF2003	94.02	94.02	91.93	91.93	95.59	93.65	93.83	93.83	95.74	95.17	94.80	94.77	95.72	95.72
CLEF2004	89.88	89.88	86.59	86.59	90.97	91.06	93.62	93.62	93.42	92.95	94.79	94.82	96.26	95.85
CLEF2005	87.98	87.98	87.34	87.41	91.35	89.15	95.90	95.90	97.92	97.52	96.31	96.37	98.19	97.25
CLEF2006	88.92	88.92	89.64	89.64	92.16	91.17	92.77	92.77	94.98	94.55	95.04	95.04	95.04	95.47
CLEF2007	92.48	92.48	91.07	91.13	95.44	93.32	94.60	94.60	95.63	95.69	94.21	94.21	95.95	95.43

Table 3: Results on the CLEF data with varying amount of training data

training data in table 3). In such cases it is more beneficial to just apply the original Alpino model or the simple model combination technique.

4.4 Experiments with CGN

One might argue that the question domain is rather 'easy', given the already high baseline performance and the fact that few hand-annotated questions are enough to obtain a reasonable model. Therefore, we examine our approach on CGN (Oostdijk, 2000).

The empirical results of testing using cross-validation within a subset of CGN subdomains are given in table 4. The baseline accuracies are much lower on this more heterogeneous, spoken, data, leaving more room for potential improvements over the in-domain model. However, the results show that the auxiliary based approach does not work on the CGN subdomains either. The approach is not able to improve even on datasets where very little training data is available (e.g. comp-l), thus confirming our previous finding. Moreover, in some cases the auxiliary feature rather, although only slightly, *degrades* performance (indicated in italic in table 4) and performs worse than the counterpart model without the additional feature.

Depending on the different characteristics of data/domain and its size, the best model adaptation method varies on CGN. On some subdomains simple model combination performs best, while on others it is more beneficial to just apply the original, out-of-domain Alpino model.

To conclude, model combination achieves in most cases a modest improvement, while we have shown empirically that our domain adaptation method based on auxiliary distributions performs just similar to a model trained on in-domain data.

5 Conclusions

We examined auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation. While the auxiliary approach has been successfully applied to lexical selectional preferences (Johnson and Riezler, 2000; van Noord, 2007), our empirical results show that integrating a more general into a domain-specific model through the auxiliary feature approach does not help. The auxiliary approach needs training data to estimate the weight(s) of the auxiliary feature(s). When little training data is available, the weight cannot be estimated appropriately and hence is not contributing enough compared to the other features. This result was confirmed on both examined domains. We conclude that the auxiliary feature approach is not appropriate for integrating information of a more general model to leverage limited in-domain data. Better results were achieved either without adaptation or by simple model combination.

Future work will consist in investigating other possibilities for parser adaptation, especially *semi-supervised* domain adaptation, where no labeled in-domain data is available.

References

Abney, Steven P. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23:597–618.

Berger, A. and H. Printz. 1998. A comparison of criteria for maximum entropy / minimum divergence feature selection. In *In Proceedings of the 3nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 97–106, Granada, Spain.

Berger, Adam, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72.

Boros, M., W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. 1996. Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In *Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP 96)*, Philadelphia.

Briscoe, Ted, John Carroll, Jonathan Graham, and Ann Copestake. 2002. Relational evaluation schemes. In *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation*, pages 4–8, Las Palmas, Gran Canaria.

Gildea, Daniel. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Hara, Tadayoshi, Miyao Yusuke, and Jun'ichi Tsujii. 2005. Adapting a probabilistic disambiguation model of an hpsg

comp-a (1,193) - Spontaneous conversations (’face-to-face’)						comp-b (525) - Interviews with teachers of Dutch					
DataSet	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb. eq.weights	Dataset	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb eq.weights
fn000250	63.20	63.28	62.90	63.91	**63.99**	fn000081	66.20	66.39	66.45	**67.26**	66.85
fn000252	64.74	64.74	64.06	64.87	**64.96**	fn000089	62.41	62.41	63.88	**64.35**	64.01
fn000254	66.03	*66.00*	65.78	66.39	**66.44**	fn000086	62.60	62.76	63.17	63.59	**63.77**
comp-l (116) - Commentaries/columns/reviews (broadcast)						comp-m (267) - Ceremonious speeches/sermons					
DataSet	no aux	+ aux	Alpino	Mod.Comb.	Model.Comb. eq.weights	Dataset	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb eq.weights
fn000002	67.63	67.63	**77.30**	76.96	72.40	fn000271	59.25	59.25	63.78	**64.94**	61.76
fn000017	64.51	*64.33*	**66.42**	66.30	65.74	fn000298	70.33	*70.19*	74.55	**74.83**	72.70
fn000021	61.54	61.54	**64.30**	64.10	63.24	fn000781	72.26	72.37	**73.55**	**73.55**	73.04

Table 4: Excerpt of results on various CGN subdomains (# of sentences in parenthesis).

parser to a new domain. In *Proceedings of the International Joint Conference on Natural Language Processing*.

Johnson, Mark and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 154–161, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic “unification-based” grammars. In *Proceedings of the 37th Annual Meeting of the ACL*.

Lease, Matthew, Eugene Charniak, Mark Johnson, and David McClosky. 2006. A look at parsing and its applications. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 16–20 July.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

Oostdijk, Nelleke. 2000. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC)*, pages 887–894.

Osborne, Miles. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000)*.

Ratnaparkhi, A. 1997. A simple introduction to maximum entropy models for natural language processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania.

van Noord, Gertjan and Robert Malouf. 2005. Wide coverage parsing with stochastic attribute value grammars. Draft available from http://www.let.rug.nl/˜vannoord. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan China, 2004.

van Noord, Gertjan. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

van Noord, Gertjan. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the Tenth International Conference on Parsing Technologies. IWPT 2007, Prague.*, pages 1–10, Prague.

van Noord, Gertjan. In preparation. Learning efficient parsing.

Velldal, E. and S. Oepen. 2005. Maximum entropy models for realization ranking. In *Proceedings of MT-Summit*, Phuket, Thailand.

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/

Proceedings of the Coling 2008 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 linguistic 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.

Toward a cross-framework parser annotation standard

Dan Flickinger
CSLI, Stanford University
danf@stanford.edu

Abstract

Efficient and precise comparison of parser results across frameworks will require a negotiated agreement on a target representation which embodies a good balance of three competing dimensions: consistency, clarity, and flexibility. The various annotations provided in the COLING-08 shared task for the ten 'required' Wall Street Journal sentences can serve as a useful basis for these negotations. While there is of course substantial overlap in the content of the various schemes for these sentences, no one of the schemes is ideal. This paper presents some desiderata for a negotiated target annotation scheme for which straightforward mappings can be constructed from each of the supplied annotation schemes.

1 Introduction

Efficient and precise comparison of parser results across frameworks will require a negotiated agreement on a target representation which embodies a good balance of three competing dimensions: consistency, clarity, and flexibility. The various annotations provided in the COLING-08 shared task for the ten 'required' Wall Street Journal sentences can serve as a useful basis for these negotations. While there is of course substantial overlap in the content of the various schemes for these sentences, no one of the schemes is ideal, containing either too much or too little detail, or sometimes both.

2 Predicate-argument structures, not labelled bracketings

Competing linguistic frameworks can vary dramatically in the syntactic structures they assign to sentences, and this variation makes cross-framework comparison of labelled bracketings difficult and in the limit uninteresting. The syntactic structures of Combinatory Categorial Grammar (CCG: Steedman (2000), Hockenmaier (2003), Clark and Curran (2003)), for example, contrast sharply with those of the Penn Treebank Marcus et al. (1993), and the PTB structures differ in many less dramatic though equally important details from those assigned in Lexical Functional Grammar (LFG: Bresnan and Kaplan (1982)) or Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag (1994)). We even find variation in the assignments of part-of-speech tags for individual tokens, for example with words like "missionary" or "classical" treated as adjectives in some of the annotations and as nouns in others. Furthermore, a simple labelled bracketing of surface tokens obscures the fact that a single syntactic constituent can fill multiple roles in the logical structure expressed by a sentence, as with controlled subjects, relative clauses, appositives, coordination, etc. More detailed discussions of the obstacles to directly comparing syntactic structures include Preiss (2003), Clark and Curran (2007), and most recently Sagae et al. (2008).

Since it is this underlying logical content that we seek when parsing a sentence, the target annotation for cross-framework comparison should not include marking of syntactic constituents, but focus instead on the predicate argument structures determined by the syntactic analysis, as proposed ten years ago by Carroll et al. (1998). Several of the annotations provided in the shared task already do this, providing a good set of starting points for negotiating a common target.

3 General annotation characteristics

Some of the issues in need of negotiation are quite general in nature, while many others involve specific phenomena. First, the general ones:

3.1 Unique identifiers

Since a given word can appear multiple times within a single sentence, each token-derived element of the annotation needs a unique identifier. Some of the supplied annotations use the token position in the sentence for this purpose, but this is not general enough to support competing hypotheses about the number of tokens in a sentence. A sharp example of this is the word *pixie-like* in sentence 56, which one of the annotations (CONLL08) analyzes as two tokens, quite reasonably, since *-like* is a fully productive compounding element. So a better candidate for the unique identifier for each annotation element would be the initial *character* position of the source token in the original sentence, including spaces and punctuation marks as characters. Thus in the sentence *the dog slept* the annotation elements would be *the-1*, *dog-5*, and *slept-9*. The original sentences in this shared task were presented with spaces added around punctuation, and before "n't". so the character positions for this task would be computed taking this input as given. Using character positions rather than token positions would also better accommodate differing treatments of multi-word expressions, as for example with *Los Angeles* in sentence 9, which most of the supplied schemes annotate as two tokens with *Los* modifying *Angeles*, but which PARC treats as a single entity.

3.2 One token in multiple roles

Most of the supplied annotations include some notational convention to record the fact that (a phrase headed by) a single token can fill more than one logical role at the predicate-argument level of representation. This is clear for controlled subjects as in the one for *play* in sentence 53: *"doesn't have to play...concertos"*, and equally clear for the missing objects in *tough*-type adjective phrases, like the object of *apply* in sentence 133: *"impossible to apply"*. This multiple filling of roles by a single syntactic constituent can be readily expressed in a target annotation of the predicate argument structure if the token heading that constituent bears the unique positional identifier which has already been motivated above. Supplied annotation schemes that already directly employ this approach include PARC and Stanford, and the necessary positional information is also readily available in the CCG-PA, HPSG-PA, and CONLL08 schemes, though not in the RASP-GR or PTB notations. It will be desirable to employ this same convention for the logical dependencies in other constructions with missing arguments, including relative clauses, other unbounded dependencies like questions, and comparative constructions like sentence 608's *than President Bush has allowed* __.

3.3 Stem vs surface form

Some of the supplied annotations (CCG-PA, RASP-GR, and Stanford) simply use the surface forms of the tokens as the elements of relations, while most of the others identify the stem forms for each token. While stemming might introduce an additional source of inconsistency in the annotations, the resulting annotations will be better normalized if the stems rather than the surface forms of words are used. This normalization would also open the door to making such annotations more suitable for validation by reasoning engines, or for later word-sense annotation, or for applications.

3.4 Identification of root

Most but not all of the supplied annotation schemes identify which token supplies the outermost predication for the sentence, either directly or indirectly. An explicit marking of this outermost element, typically the finite verb of the main clause of a sentence, should be included in the target annotation, since it avoids the spurious ambiguity found for example in the HPSG-PA annotation for sentence 22, which looks like it would be identical for both of the following two sentences:

- *Not all those who wrote oppose the changes* .
- *Not all those who oppose the changes wrote* .

3.5 Properties of entities and events

Some of the supplied annotation schemes include information about morphosyntactically marked properties of nouns and verbs, including person, number, gender, tense, and aspect. Providing for explicit marking of these properties in a common target annotation is desirable, at least to the level of detail adopted by several of the supplied schemes.

While several of the supplied annotation schemes marked some morphosyntactic properties some of the time, the PARC annotation of positive degree for all adjectives reminds us that it would be useful to adopt a notion of default values for these properties in the target annotation. These defaults would be explicitly defined once, and then only non-default values would need to be marked explicitly in the annotation for a given sentence. For example, the PARC annotation marks the 'perf' (perfect) attribute for a verb only when it has a positive value, implicitly using the negative value as the default. This use of defaults would improve the readability of the target annotation without any loss of information.

Marking of the contrast between declarative, interrogative, and imperative clauses is included in some but not all of the annotation schemes. Since this contrast is highly salient and (almost always) easily determined, it should be marked explicitly in the target annotation, at least for the main clause.

3.6 Named entities

The supplied annotations represent a variety of approaches to the treatment of named entities where multiple tokens comprise the relevant noun phrase, as in sentence 53's *"The oboist Heinz Holliger"*. Several schemes treat both *oboist* and *Heinz* simply as modifiers of *Holliger*, drawing no distinction between the two. The PARC and PTB annotations identify *Heinz Holliger* as a named entity, with *oboist* as a modifier, and only the CONLL08 scheme analyses this expression as an apposition, with *oboist* as the head predicate of the whole PN. Since complex proper names appear frequently with modifiers and in apposition constructions, and since competing syntactic and semantic analyses can be argued for many such constituents, the target annotation should contain enough detail to illuminate the substantive differences without exaggerating them. Interestingly, this suggests that the evaluation of a given analysis in comparison with a gold standard in the target annotation may require some computation of near-equivalence at least for entities in complex noun phrases. If scheme A treats *Holliger* as the head token for use in external dependencies involving the above noun phrase, while scheme B treats *oboist* as the head token, it will be important in evaluation to exploit the fact that both schemes each establish some relation between *oboist* and *Holliger* which can be interpreted as substitutional equivalence with respect to those external dependencies. This means that even when a target annotation scheme has been agreed upon, and a mapping defined to convert a native annotated analyis into a target annotation, it will still be necessary to create non-trivial software which can evaluate the mapped analysis against a gold standard analysis.

4 Notational conventions to be negotiated

A number of notational conventions will have to be negotiated for a common target annotation scheme, ranging from quite general design decisions to details about very specific linguistic phenomena.

4.1 Naming of arguments and relations

It seems plausible that agreement could be reached quickly on the names for at least the core grammatical functions of subject, direct object, indirect object, and verbal complement, and perhaps also on the names for adjectival and adverbial modifiers. Prepositions are more challenging, since they are very often two-place relations, and often live on the blurry border between arguments and adjuncts. For example, most of the supplied annotation schemes treated the *by*-PP following *moved* in sentence 608 as a marker for the logical subject of the passive verb, but this was at least not clear in the CCG-PA annotation. In sentence 56, there was variation in how the *from* and *to* PPs were annotated, with CONLL08 making the two *to* PPs dependents of the *from* PP rather than of the verb *range*.

Some of the supplied annotation schemes introduced reasonable but idiosyncratic names for other frequently occurring relations or dependencies such as relative clauses, appositives, noun-noun compounds, and subordinate clauses. An inventory of these frequently occurring phenomena should be constructed, and a target name negotiated for each, recognizing that there will always be a long tail of less frequently occurring phenomena where names will not (yet) have been negotiated.

4.2 Coordination

Perhaps the single most frequent source of apparent incompatibility in the supplied annotations for the ten required sentences in this task involves coordination. Some schemes, like HPSG-PA and

Stanford, treat the first conjunct as the primary entity which participates in other predications, with the other conjunct(s) dependent on the first, though even here they usually (but not always) distribute conjoined verbal arguments with separate predications for each conjunct. Some schemes, like the PTB, PARC, and RASP-GR, represent the grouping of three or more conjuncts as flat, while others like the Stanford scheme represent them as pairs. Most schemes make each conjunction word itself explicit, but for example the PARC annotation of 866 marks only one occurrence of *and* even though this three-part coordinate structure includes two explicit conjunctions.

While the distribution of conjoined elements in coordinate structures may be the most practical target annotation, it should at least be noted that this approach will not accommodate collective readings of coordinate NPs as in well-known examples like *"Tom and Mary carried the piano upstairs."* But the alternative, to introduce a new conjoined entity for every coordinate structure, may be too abstract to find common support among developers of current annotation schemes, and perhaps not worth the effort at present.

However, it should be possible to come to agreement on how to annotate the distribution of conjoined elements consistently, such that it is clear both which elements are included in a coordinate structure, and what role each plays in the relevant predicate argument structures.

4.3 Verb-particle expressions

Another phenomenon exhibited several times in these ten sentences involves verb-particle expressions, as with *thrash out* and perhaps also *stop by*. Most of the supplied schemes distinguished this dependency, but some simply treated the particle as a modifier of the verb. It would be desirable to explicitly distinguish in a target annotation the contrast between *stopped a session* and *stopped by a session* without having to hunt around in the annotation to see if there happens to be a modifier of *stop* that would dramaticaly change its meaning.

The example with *stop by a session* also highlights the need for an annotation scheme which localizes the differences between competing analyses where possible. Though all of the supplied annotations treat *by* as a particle just like *up* in *"look up the answer"*, in fact *by* fails the clearest test for being a particle, namely the ability to appear after the NP argument: *"*He stopped the session by."* An analysis treating *"by the session"* as a selected-for PP with a semantically empty *by* might better fit the linguistic facts, but the target annotation could remain neutral about this syntactic debate if it simply recorded the predicate as *stop_by*, taking an NP argument just as is usually done for the complement of *rely* in *"rely on us"*.

4.4 Less frequent phenomena

Since each new phenomenon encountered may well require negotiation in order to arrive at a common target annotation, it will be important to include some provisional annotation for relations that have not yet been negotiated. Even these ten example sentences include a few expressions where there was little or no agreement among the schemes about the annotations, such as *"if not more so"* in sentence 30, or *"to be autographed"* in sentence 216. It would be convenient if the target annotation scheme included a noncommittal representation for some parts of a given sentence explicitly noting the lack of clarity about what the structure should be.

4.5 Productive derivational morphology

It was surprising that only one of the annotation schemes (CONLL08) explicitly annotated the nominal gerund *conducting* in sentence 53 as productively related to the verb *conduct*.. While the issue of derivational morphology is of course a slippery slope, the completely productive gerund-forming process in English should be accommodated in any target annotation scheme, as should a small number of other highly productive and morphologically marked derivational regularities, including participial verbs used as prenominal modifiers, and comparative and superlative adjectives. Including this stemming would provide an informative level of detail in the target annotation, and one which can almost always be readily determined from the syntactic context.

5 Next steps

The existing annotation schemes supplied for this task exhibit substantial common ground in the nature and level of detail of information being recorded, making plausible the idea of investing a modest amount of joint effort to negotiate a common target representation which addresses at least some of the issues identified here. The initial common target annotation scheme should be one which has the following properties:

- Each existing scheme's annotations can be readily mapped to those of the target scheme via an automatic procedure.
- The annotations appear in compact, humanly readable form as sets of tuples recording either predicate-argument dependencies or properties of entities and events, such as number and tense.
- The inventory of recorded distinctions is rich enough to accommodate most of what any one scheme records, though it may not be a superset of all such distinctions. For example, some scheme might record quantifier scope information, yet the target annotation scheme might not, either because it is not of high priority for most participants, or because it would be difficult to produce consistently in a gold standard.

The primary purposes of such a target annotation scheme should be to facilitate the automatic comparison of results across frameworks, and to support evaluation of results against gold standard analyses expressed in this target scheme. It might also be possible to define the scheme such that the target annotations contain enough information to serve as the basis for some application-level tasks such as reasoning, but the primary design criteria should be to enable detailed comparison of analyses.

References

Bresnan, Joan and Ronald M. Kaplan, 1982. Lexical-Functional Grammar. A Formal System for Grammatical Representation. *The Mental Representation of Grammatical Relations,* ed. Joan Bresnan. MIT Press, Cambridge, MA.

Carroll, John, Edward Briscoe and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. *Proceedings of the 1st International Conference on Language Resources and Evaluation.*

Clark, Stephen and James R. Curran. 2003. Log-linear models for wide-coverage CCG parsing. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp.97–104.

Clark, Stephen and James R. Curran. 2007. Formalism-Independent Parser Evaluation with CCG and DepBank. *Proceedings of the Association for Computational Linguistics 2007.*

Harrison, P., S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, D. Hindle, B. Ingria, M. Marcus, B. Santorini, , and T. Strzalkowski. 1991. Evaluating syntax performance of parser/grammars of English. *Natural Language Processing Systems Evaluation Workshop,* Technical Report RL- TR-91-6, J. G. Neal and S. M. Walter, eds.

Hockenmaier, Julia. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar.* Ph.D. thesis, University of Edinburgh.

Marcus, Mitchell P., Beatrice Santorini and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics* 19:313–330.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Preiss, Judita. 2003. Using Grammatical Relations to Compare Parsers. *Proceedings of the European Association for Computational Linguistics 2003.*

Sagae, Kenji, Yusuke Miyao, Takuya Matsuzaki and Jun'ichi Tsujii. 2008. Challenges in Mapping of Syntactic Representations for Framework-Independent Parser Evaluation. *Proceedings of the Workshop on Automated Syntatic Annotations for Interoperable Language Resources at the 1st International Conference on Global Interoperability for Language Resources*, pp.61–68.

Steedman, Mark. 2000. *The syntactic process.* MIT Press, Cambridge, MA.

Parser Evaluation across Frameworks without Format Conversion

Wai Lok Tam
Interfaculty Initiative in
Information Studies
University of Tokyo
7-3-1 Hongo Bunkyo-ku
Tokyo 113-0033 Japan

Yo Sato
Dept of Computer Science
Queen Mary
University of London
Mile End Road
London E1 4NS, U.K.

Yusuke Miyao **Jun-ichi Tsujii**
Dept of Computer Science
University of Tokyo
7-3-1 Hongo Bunkyo-ku
Tokyo 113-0033 Japan

Abstract

In the area of parser evaluation, formats like GR and SD which are based on dependencies, the simplest representation of syntactic information, are proposed as framework-independent metrics for parser evaluation. The assumption behind these proposals is that the simplicity of dependencies would make conversion from syntactic structures and semantic representations used in other formalisms to GR/SD a easy job. But (Miyao et al., 2007) reports that even conversion between these two formats is not easy at all. Not to mention that the 80% success rate of conversion is not meaningful for parsers that boast 90% accuracy. In this paper, we make an attempt at evaluation across frameworks without format conversion. This is achieved by generating a list of names of phenomena with each parse. These names of phenomena are matched against the phenomena given in the gold standard. The number of matches found is used for evaluating the parser that produces the parses. The evaluation method is more effective than evaluation methods which involve format conversion because the generation of names of phenomena from the output of a parser loaded is done by a recognizer that has a 100% success rate of recognizing a phenomenon illustrated by a sentence. The success rate is made possible by the reuse of native codes: codes used for writing the parser and rules of the grammar loaded into the parser.

1 Introduction

The traditional evaluation method for a deep parser is to test it against a list of sentences, each of which is paired with a yes or no. The parser is evaluated on the number of grammatical sentences it accepts and that of ungrammatical sentences it rules out. A problem with this approach to evaluation is that it neither penalizes a parser for getting an analysis wrong for a sentence nor rewards it for getting it right. What prevents the NLP community from working out a universally applicable reward and penalty scheme is the absence of a gold standard that can be used across frameworks. The correctness of an analysis produced by a parser can only be judged by matching it to the analysis produced by linguists in syntactic structures and semantic representations created specifically for the framework on which the grammar is based. A match or a mismatch between analyses produced by different parsers based on different frameworks does not lend itself for a meaningful comparison that leads to a fair evaluation of the parsers. To evaluate two parsers across frameworks, two kinds of methods suggest themselves:

1. Converting an analysis given in a certain format native to one framework to another native to a differernt framework (e.g. converting from a CCG (Steedman, 2000) derivation tree to an HPSG (Pollard and Sag, 1994) phrase structure tree with AVM)

2. Converting analyses given in different framework-specific formats to some simpler format proposed as a framework-independent evaluation schema (e.g. converting from

Proceedings of the Coling 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 29–35
Manchester, August 2008

HPSG phrase structure tree with AVM to GR (Briscoe et al., 2006))

However, the feasibility of either solution is questionable. Even conversion between two evaluation schemata which make use of the simplest representation of syntactic information in the form of dependencies is reported to be problematic by (Miyao et al., 2007).

In this paper, therefore, we propose a different method of parser evaluation that makes no attempt at any conversion of syntactic structures and semantic representations. We remove the need for such conversion by abstracting away from comparison of syntactic structures and semantic representations. The basic idea is to generate a list of names of phenomena with each parse. These names of phenomena are matched against the phenomena given in the gold standard for the same sentence. The number of matches found is used for evaluating the parser that produces the parse.

2 Research Problem

Grammar formalisms differ in many aspects. In syntax, they differ in POS label assignment, phrase structure (if any), syntactic head assignment (if any) and so on, while in semantics, they differ from each other in semantic head assignment, role assignment, number of arguments taken by predicates, etc. Finding a common denominator between grammar formalisms in full and complex representation of syntactic information and semantic information has been generally considered by the NLP community to be an unrealistic task, although some serious attempts have been made recently to offer simpler representation of syntactic information (Briscoe et al., 2006; de Marneffe et al., 2006).

Briscoe et al (2006)'s Grammatical Relation (GR) scheme is proposed as a framework-independent metric for parsing accuracy. The promise of GR lies actually in its dependence on a framework that makes use of simple representation of syntactic information. The assumption behind the usefulness of GR for evaluating the output of parsers is that most conflicts between grammar formalisms would be removed by discarding less useful information carried by complex syntactic or semantic representations used in grammar formalisms during conversion to GRs. But is this assumption true? The answer is not clear. A GR represents syntactic information in the form of a binary relation between a token assigned as the head of the relation and other tokens assigned as its dependents. Notice however that grammar frameworks considerably disagree in the way they assign heads and non-heads. This would raise the doubt that, no matter how much information is removed, there could still remain disagreements between grammar formalisms in what is left.

The simplicity of GR, or other dependency-based metrics, may give the impression that conversion from a more complex representation into it is easier than conversion between two complex representations. In other words, GRs or a similar dependency relation looks like a promising candidate for *lingua franca* of grammar frameworks. However the experiment results given by Miyao et al (2007) show that even conversion into GRs of predicate-argument structures, which is not much more complex than GRs, is not a trivial task. Miyao et al (2007) manage to convert 80% of the predicate-argument structures outputted by their deep parser, ENJU, to GRs correctly. However the parser, with an over 90% accuracy, is too good for the 80% conversion rate. The lesson here is that simplicity of a representation is a different thing from simplicity in converting into that representation.

3 Outline of our Solution

The problem of finding a common denominator for grammar formalisms and the problem of conversion to a common denominator may be best addressed by evaluating parsers without making any attempt to find a common denominator or conduct any conversion. Let us describe briefly in this section how such evaluation can be realised.

3.1 Creating the Gold Standard

The first step of our evaluation method is to construct or find a number of sentences and get an annotator to mark each sentence for the phenomena illustrated by each sentence. After annotating all the sentences in a test suite, we get a list of pairs, whose first element is a sentence ID and second is again a list, one of the corresponding phenomena. This list of pairs is our gold standard. To illustrate, suppose we only get sentence 1 and sentence 2 in our test suite.

(1) John gives a flower to Mary

(2) John gives Mary a flower

Sentence 1 is assigned the phenomena: proper noun, unshifted ditransitive, preposition. Sentence 2 is assigned the phenomena: proper noun, dative-shifted ditransitive. Our gold standard is thus the following list of pairs:

$\langle 1, \langle$proper noun, unshifted ditransitive, preposition$\rangle \rangle$,
$\langle 2, \langle$proper noun,dative-shifted ditransitive$\rangle \rangle$

3.2 Phenomena Recognition

The second step of our evaluation method requires a small program that recognises what phenomena are illustrated by an input sentence taken from the test suite based on the output resulted from parsing the sentence. The recogniser provides a set of conditions that assign names of phenomena to an output, based on which the output is matched with some framework-specific regular expressions. It looks for hints like the rule being applied at a node, the POS label being assigned to a node, the phrase structure and the role assigned to a reference marker. The names of phenomena assigned to a sentence are stored in a list. The list of phenomena forms a pair with the ID of the sentence, and running the recogniser on multiple outputs obtained by batch parsing (with the parser to be evaluated) will produce a list of such pairs, in exactly the same format as our gold standard. Let us illustrate this with a parser that:

1. assigns a monotransitive verb analysis to 'give' and an adjunct analysis to 'to Mary' in 1

2. assigns a ditransitive verb analysis to 'give' in 2

The list of pairs we obtain from running the recogniser on the results produced by batch parsing the test suite with the parser to be evaluated is the following:

$\langle 1, \langle$proper noun,monotransitive,preposition,adjunct$\rangle \rangle$,
$\langle 2, \langle$proper noun,dative-shifted ditransitive$\rangle \rangle$

3.3 Performance Measure Calculation

Comparing the two list of pairs generated from the previous steps, we can calculate the precision and recall of a parser using the following formulae:

$$Precision = (\sum_{i=1}^{n} \frac{\mid R_i \cap A_i \mid}{\mid R_i \mid}) \div n \qquad (1)$$

$$Recall = (\sum_{i=1}^{n} \frac{\mid R_i \cap A_i \mid}{\mid A_i \mid}) \div n \qquad (2)$$

where list R_i is the list generated by the recogniser for sentence i, list A_i is the list produced by annotators for sentence i, and n the number of sentences in the test suite.

In our example, the parser that does a good job with dative-shifted ditransitives but does a poor job with unshifted ditranstives would have a precision of:

$$(\frac{2}{4} + \frac{2}{2}) \div 2 = 0.75$$

and a recall of:

$$(\frac{2}{3} + \frac{2}{2}) \div 2 = 0.83$$

4 Refining our Solution

In order for the precision and recall given above to be a fair measure, it is necessary for both the recogniser and the annotators to produce an exhaustive list of the phenomena illustrated by a sentence.

But we foresee that annotation errors are likely to be a problem of exhaustive annotation, as is reported in Miyao et al (2007) for the gold standard described in Briscoe et al (2006). Exhaustive annotation procedures require annotators to repeatedly parse a sentence in search for a number of phenomena, which is not the way language is normally processed by humans. Forcing annotators to do this, particularly for a long and complex sentence, is a probable reason for the annotation errors in the gold standard described in (Briscoe et al., 2006).

To avoid the same problem in our creation of a gold standard, we propose to allow non-exhaustive annotation. In fact, our proposal is to limit the number of phenomena assigned to a sentence to one. This decision on which phenomenon to be assigned is made, when the test suite is constructed, for each of the sentences contained in it. Following the traditional approach, we include every sentence in the test suite, along with the core phenomenon we intend to test it on (Lehmann and Oepen, 1996). Thus, Sentence 1 would be assigned the phenomenon of unshifted ditransitive. Sentence 2 would be assigned the phenomenon of dative-shifted ditransitive. This revision of annotation policy removes the need for exhaustive annotation. Instead, annotators are given a new task. They are asked to assign to each sentence *the most common error that a parser is likely to make*. Thus Sentence 1 would be assigned adjunct for such an error. Sentence 2 would be assigned the error of noun-noun compound. Note that these errors are also names of phenomena.

This change in annotation policy calls for a change in the calculation of precision and recall. We leave the recogniser as it is, i.e. to produce an exhaustive list of phenomena, since it is far beyond our remit to render it intelligent enough to select a single, intended, phenomenon. Therefore, an incorrectly low precision would result from a mismatch between the exhaustive list generated by the recogniser and the singleton list produced by annotators for a sentence. For example, suppose we only have sentence 2 in our test suite and the parser correctly analyses the sentence. Our recogniser assigns two phenomena (proper noun, dative-shifted ditransitive) to this sentence as before. This would result in a precision of 0.5.

Thus we need to revise our definition of precision, but before we give our new definition, let us define a truth function t:

$$t(A \supset B) = \begin{cases} 1 & A \quad \supset B \\ 0 & A \cap B \quad = \emptyset \end{cases}$$

$$t(A \cap B = \emptyset) = \begin{cases} 0 & A \cap B \quad \neq \emptyset \\ 1 & A \cap B \quad = \emptyset \end{cases}$$

Now, our new definition of precision and recall is as follows:

$$Precision \qquad (3)$$
$$= \frac{(\sum_{i=1}^{n} \frac{t(R_i \supset AP_i) + t(R_i \cap AN_i = \emptyset)}{2})}{n}$$

$$Recall \qquad (4)$$
$$= \frac{(\sum_{i=1}^{n} \frac{|R_i \cap AP_i|}{|AP_i|})}{n}$$

where list AP_i is the list of phenomena produced by annotators for sentence i, and list AN_i is the list of errors produced by annotators for sentence i.

While the change in the definition of recall is trivial, the new definition of precision requires some explanation. The exhaustive list of phenomena generated by our recogniser for each sentence is taken as a combination of two answers to two questions on the two lists produced by annotators for each sentence. The correct answer to the question on the one-item-list of phenomenon produced by annotators for a sentence is a superset-subset relation between the list generated by our recogniser and the one-item-list of phenomenon produced by annotators. The correct answer to the question on the one-item-list of error produced by annotators for a sentence is the non-existence of any common member between the list generated by our recogniser and the one-item-list of error produced by annotators.

To illustrate, let us try a parser that does a good job with dative-shifted ditransitives but does a poor job with unshifted ditranstives on both 2 and 1. The precision of such a parser would be:

$$(\frac{0}{2} + \frac{2}{2}) \div 2 = 0.5$$

and its recall would be:

$$(\frac{0}{1} + \frac{1}{1}) \div 2 = 0.5$$

5 Experiment

For this abstract, we evaluate ENJU (Miyao, 2006), a released deep parser based on the HPSG formalism and a parser based on the Dynamic Syntax formalism (Kempson et al., 2001) under development against the gold standard given in table 1.

The precision and recall of the two parsers (ENJU and DSPD, which stands for ”Dynamic Syntax Parser under Development”) are given in table 3:

The experiment that we report here is intended to be an experiment with the evaluation method described in the last section, rather than a very serious attempt to evaluate the two parsers in question. The sentences in table 1 are carefully selected to include both sentences that illustrate core phenomena and sentences that illustrate rarer but more interesting (to linguists) phenomena. But there are too few of them. In fact, the most important number that we have obtained from our experiment is the 100% success rate in recognizing the phenomena given in table 1.

ID	Phenomenon	Error
1	unshifted ditransitive	adjunct
2	dative-shifted ditransitive	noun-noun compound
3	passive	adjunct
4	nominal gerund	verb that takes verbal complement
5	verbal gerund	imperative
6	preposition	particle
7	particle	preposition
8	adjective with extrapolated sentential complement	relative clause
9	inversion	question
10	raising	control

Figure 1: Gold Standard for Parser Evaluation

ID	Sentence
1	John gives a flower to Mary
2	John give Mary a flower
3	John is dumped by Mary
4	Your walking me pleases me
5	Abandoning children increased
6	He talks to Mary
7	John makes up the story
8	It is obvious that John is a fool
9	Hardly does anyone know Mary
10	John continues to please Mary

Figure 2: Sentences Used in the Gold Standard

Measure	ENJU	DSPD
Precision	0.8	0.7
Recall	0.7	0.5

Figure 3: Performance of Two Parsers

6 Discussion

6.1 Recognition Rate

The 100% success rate is not as surprising as it may look. We made use of two recognisers, one for each parser. Each of them is written by the one of us who is somehow involved in the development of the parser whose output is being recognised and familiar with the formalism on which the output is based. This is a clear advantage to format conversion used in other evaluation methods, which is usually done by someone familiar with either the source or the target of conversion, but not both, as such a recogniser only requires knowledge of one formalism and one parser. For someone who is involved in the development of the grammar and of the parser that runs it, it is straightforward to write a recogniser that can make use of the code built into the parser or rules included in the grammar. We can imagine that the 100% recognition rate would drop a little if we needed to recognise a large number of sentences but were not allowed sufficient time to write detailed regular expressions. Even in such a situation, we are confident that the success rate of recognition would be higher than the conversion method.

Note that the effectiveness of our evaluation method depends on the success rate of recognition to the same extent that the conversion method employed in Briscoe et al. (2006) and de Marneff et al. (2006) depends on the conversion rate. Given the high success rate of recognition, we argue that our evaluation method is more effective than any evaluation method which makes use of a format claimed to be framework independent and involves conversion of output based on a different formalism to the proposed format.

6.2 Strictness of Recognition and Precision

There are some precautions regarding the use of our evaluation method. The redefined precision 4 is affected by the strictness of the recogniser. To illustrate, let us take Sentence 8 in Table 1 as an example. ENJU provides the correct phrase structure analysis using the desired rules for this sentence but makes some mistakes in assigning roles to the adjective and the copular verb. The recogniser we write for ENJU is very strict and refuses to assign the phenomenon ‘adjective with extrapolated sentential complement’ based on the output given by ENJU. So ENJU gets 0 point for its answer to the question on the singleton list of phenomenon in the gold standard. But it gets 1 point for its answer to the question on the singleton list of error in the gold standard because it does not go to the other extreme: a relative clause analysis, yielding a 0.5 precision. In this case, this value is fair for ENJU, which produces a partially correct analysis. However, a parser that does not accept the sentence at all, a parser that fails to produce any output or one that erroneously produces an unexpected phenomenon would get the same result: for Sentence 8, such a parser would still get a precision of 0.5, simply because its output does not show that it assigns a relative clause analysis.

We can however rectify this situation. For the lack of parse output, we can add an exception clause to make the parser automatically get a 0 precision (for that sentence). Parsers that make unexpected mistakes are more problematic. An obvious solution to deal with these parsers is to come up with an exhaustive list of mistakes but this is an unrealistic task. For the moment, a temporary but realistic solution would be to expand the list of errors assigned to each sentence in the gold standard and ask annotators to make more intelligent guess of the mistakes that can be made by parsers by considering factors such as similarities in phrase structures or the sharing of sub-trees.

6.3 Combining Evaluation Methods

For all measures, some distortion is unavoidable when applied to exceptional cases. This is true for the classical precision and recall, and our redefined precision and recall is no exception. In the case of the classical precision and recall, the distortion is countered by the inverse relation between them so that even if one is distorted, we can tell from the other that how well (poorly) the object of evaluation performs. Our redefined precision and recall works pretty much the same way.

What motivates us to derive measures so closely related to the classical precision and recall is the ease to combine the redefined precision and recall obtained from our evaluation method with the classical precision and recall obtained from other evaluation methods, so as to obtain a full picture of the performance of the object of evaluation. For example, our redefined precision and recall figures given in Table 3 (or figures obtained from running the same experiment on a larger test set) for ENJU can be combined with the precision and recall figures given in Miyao et al. (2006) for ENJU, which is based on a evaluation method that compares its predicate-argument structures those given in Penn Treebank. Here the precision and recall figures are calculated by assigning an equal weight to every sentence in Section 23 of Penn Treebank. This means that different weights are assigned to different phenomena depending on their frequency in the Penn Treebank. Such assignment of weights may not be desirable for linguists or developers of NLP systems who are targeting a corpus with a very different distribution of phenomena from this particular section of the Penn Treebank. For example, a linguist may wish to assign an equal weight across phenomena or more weights to ‘interesting’ phenomena. A developer of a question-answering system may wish to give more weights to question-related phenomena than other phenomena of less interest which are nevertheless attested more frequently in the Penn Treebank.

In sum, the classical precision and recall figures calculated by assigning equal weight to every sentence could be considered skewed from the perspective of phenomena, whereas our redefined precision and recall figures may be seen as skewed from the frequency perspective. Frequency is relative to domains: less common phenomena in some domains could occur more often in others. Our redefined precision and recall are not only useful for those who want a performance measure skewed the way they want, but also useful for those who want a performance measure as ‘unskewed’ as possible. This may be obtained by combining our redefined precision and recall with the classical precision and recall yielded from other evaluation methods.

7 Conclusion

We have presented a parser evaluation method that addresses the problem of conversion between frameworks by totally removing the need for that kind of conversion. We do some conversion but it is a different sort. We convert the output of a parser to a list of names of phenomena by drawing only on the framework that the parser is based on. It may be inevitable for some loss or inaccuracy to occur during this kind of intra-framework conversion if we try our method on a much larger test set with a much larger variety of longer sentences. But we are confident that the loss would still be far less than any inter-framework conversion work done in other proposals of cross-framework evaluation methods. What we believe to be a more problematic area is the annotation methods we have suggested. At the time we write this paper based on a small-scale experiment, we get slightly better result by asking our annotator to give one phenomenon and one common mistake for each sentence. This may be attributed to the fact that he is a member of the NLP community and hence he gets the knowledge to identify the core phenomena we want to test and the common error that parsers tend to make. If we expand our test set and includes longer sentences, annotators would make more mistakes whether they attempt exhaustive annotation or non-exhaustive annotation. It is difficult to tell whether exhaustive annotation or non-exhaustive annotation would be better for large scale experiments. As future work, we intend to try our evaluation method on more test data to determine which one is better and find ways to improve the one we believe to be better for large scale evaluation.

References

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of COLING/ACL 2006*.

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

Kempson, Ruth, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Lehmann, Sabine and Stephan Oepen. 1996. TSNLP test suites for natural language processing. In *Proceedings of COLING 1996*.

Miyao, Yusuke, Kenji Sagae, and Junichi Tsujii. 2007. Towards framework-independent evaluation of deep linguistic parsers. In *Proceedings of GEAF 2007*.

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

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications.

Steedman, Mark. 2000. *Syntactic Process*. MIT Press.

Large Scale Production of Syntactic Annotations to Move Forward

Patrick Paroubek, Anne Vilnat, Sylvain Loiseau
LIMSI-CNRS
BP 133 91403 Orsay Cedex
France
prenom.nom@limsi.fr

Gil Francopoulo
Tagmatica
126 rue de Picpus 75012 Paris
France
gil.francopoulo@tagmatica.com

Olivier Hamon
ELDA and LIPN-P13
55-57 rue Brillat-Savarin 75013 Paris,
France
hamon@elda.org

Eric Villemonte de la Clergerie
Alpage-INRIA
Dom. de Voluceau Rocquencourt,
B.P. 105, 78153 Le Chesnay, France
Eric.De_La_Clergerie@inria.fr

Abstract

This article presents the methodology of the PASSAGE project, aiming at syntactically annotating large corpora by composing annotations. It introduces the annotation format and the syntactic annotation specifications. It describes an important component of the methodolgy, namely an WEB-based evaluation service, deployed in the context of the first PASSAGE parser evaluation campaign.

1 Introduction

The last decade has seen, at the international level, the emergence of a very strong trend of researches on statistical methods in Natural Language Processing. In our opinion, one of its origins, in particular for English, is the availability of large annotated corpora, such as the Penn Treebank (1M words extracted from the Wall Street journal, with syntactic annotations; 2$^{\text{nd}}$ release in 1995[1], the British National Corpus (100M words covering various styles annotated with parts of speech[2]), or the Brown Corpus (1M words with morpho-syntactic annotations). Such annotated corpora were very valuable to extract stochastic grammars or to parametrize disambiguation algorithms. For instance (Miyao et al., 2004) report an experiment where an HPSG grammar is semi-automatically aquired from the Penn Treebank, by first annotating the treebank with partially specified derivation trees using heuristic rules , then by extracting lexical entries with the application of inverse grammar rules. (Cahill et al., 2004) managed to extract LFG subcategorisation frames and paths linking long distance dependencies reentrancies from f-structures generated automatically for the Penn-II treebank trees and used them in an long distance dependency resolution algorithm to parse new text. They achieved around 80% f-score for fstructures parsing on the WSJ part of the Penn-II treebank, a score comparable to the ones of the state-of-the-art hand-crafted grammars. With similar results, (Hockenmaier and Steedman, 2007) translated the Penn Treebank into a corpus of Combinatory Categorial Grammar (CCG) derivations augmented with local and long-range word to word dependencies and used it to train wide-coverage statistical parsers. The development of the Penn Treebank have led to many similar proposals of corpus annotations[3]. However, the development of such treebanks is very costly from an human point of view and represents a long standing effort, in particular for getting of rid of the annotation errors or inconsistencies, unavoidable for any kind of human annotation. Despite the growing number of annotated corpora, the volume of data that can be manually annotated remains limited thus restricting the experiments that can be tried on automatic grammar acquisition. Furthermore, designing an annotated corpus involves choices that may block future experiments from acquiring new kinds of linguistic knowledge because they necessitate annotation incompatible or difficult to produce from the existing ones.

With PASSAGE (de la Clergerie et al., 2008b), we believe that a new option becomes possible.

[1] http://www.cis.upenn.edu/~treebank/

[2] http://www.natcorp.ox.ac.uk/

[3] http://www.ims.uni-stuttgart.de/projekte/TIGER/related/links.shtml

Proceedings of the Coling 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 36–43
Manchester, August 2008

Funded by the French ANR program on Data Warehouses and Knowledge, PASSAGE is a 3-year project (2007–2009), coordinated by INRIA project-team Alpage. It builds up on the results of the EASy French parsing evaluation campaign, funded by the French Technolangue program, which has shown that French parsing systems are now available, ranging from shallow to deep parsing. Some of these systems were neither based on statistics, nor extracted from a treebank. While needing to be improved in robustness, coverage, and accuracy, these systems has nevertheless proved the feasibility to parse medium amount of data (1M words). Preliminary experiments made by some of the participants with deep parsers (Sagot and Boullier, 2006) indicate that processing more than 10 M words is not a problem, especially by relying on clusters of machines. These figures can even be increased for shallow parsers. In other words, there now exists several French parsing systems that could parse (and re-parse if needed) large corpora between 10 to 100 M words.

Passage aims at pursuing and extending the line of research initiated by the EASy campaign by using jointly 10 of the parsing systems that have participated to EASy. They will be used to parse and re-parse a French corpus of more than 100 M words along the following feedback loop between parsing and resource creation as follows (de la Clergerie et al., 2008a):

1. Parsing creates syntactic annotations;
2. Syntactic annotations create or enrich linguistic resources such as lexicons, grammars or annotated corpora;
3. Linguistic resources created or enriched on the basis of the syntactic annotations are then integrated into the existing parsers;
4. The enriched parsers are used to create richer (e.g., syntactico-semantic) annotations;
5. etc. going back to step 1

In order to improve the set of parameters of the parse combination algorithm (inspired from the Recognizer Output Voting Error Reduction, i.e. ROVER, experiments), two parsing evaluation campaigns are planned during PASSAGE, the first of these already took place at the end of 2007 (de la Clergerie et al., 2008b). In the following, we present the annotation format specification and the syntactic annotation specifications of PASSAGE, then give an account of how the syntactic annotations were compared in the first evaluation campaign, by first describing the evaluation metrics and the web server infrastructure that was deployed to process them. We conclude by showing how the results so far achieved in PASSAGE will contribute to the second part of the project, extracting and refining enriched linguistic annotations.

2 PASSAGE Annotation Format

The aim is to allow an explicit representation of syntactic annotations for French, whether such annotations come from human annotators or parsers. The representation format is intended to be used both in the evaluation of different parsers, so the parses' representations should be easily comparable, and in the construction of a large scale annotation treebank which requires that all French constructions can be represented with enough details.

The format is based on three distinct specifications and requirements:

1. MAF (ISO 24611)[4] and SynAF (ISO 24615)[5] which are the ISO TC37 specifications for morpho-syntactic and syntactic annotation (Ide and Romary, 2002) (Declerck, 2006) (Francopoulo, 2008). Let us note that these specifications cannot be called "standards" because they are work in progress and these documents do not yet have the status Published Standard. Currently, their official status is only Committee Draft.
2. The format used during the previous TECHNOLANGUE/EASY evaluation campaign in order to minimize porting effort for the existing tools and corpora.
3. The degree of legibility of the XML tagging.

From a technical point of view, the format is a compromise between "standoff" and "embedded" notation. The fine grain level of tokens and words is standoff (wrt the primary document) but higher levels use embedded annotations. A standoff notation is usually considered more powerful but less

[4] http://lirics.loria.fr/doc_pub/maf.pdf

[5] http://lirics.loria.fr/doc_pub/N421_SynAF_CD_ISO_24615.pdf

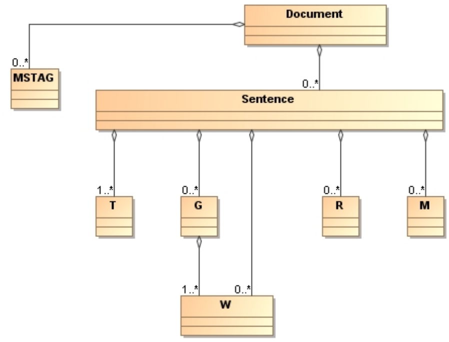


Figure 1: UML diagram of the structure of an annotated document

readable and not needed when the annotations follow a (unambiguous) tree-like structure. Let us add that, at all levels, great care has been taken to ensure that the format is mappable onto MAF and SynAF, which are basically standoff notations.

The structure of a PASSAGE annotated document may be summarized with the UML diagram in Figure1. The document begins by the declaration of all the morpho-syntactic tagsets (MSTAG) that will be used within the document. These declarations respect the ISO Standard Feature Structure Representation (ISO 24610-1). Then, tokens are declared. They are the smallest unit addressable by other annotations. A token is unsplittable and holds an identifier, a character range, and a content made of the original character string. A word form is an element referencing one or several tokens. It has has two mandatory attributes: an identifier and a list of tokens. Some optional attributes are allowed like a part of speech, a lemma, an inflected form (possibly after spelling correction or case normalization) and morpho-syntactic tags. The following XML fragment shows how the original fragment *"Les chaises"* can be represented with all the optional attributes offered by the PASSAGE annotation format :

```
<T id="t0" start="0" end="3">
   Les
</T>
<W id="w0" tokens="t0"
   pos="definiteArticle"
   lemma="le"
   form="les"
   mstag="nP"/>
<T id="t1" start="4" end="11">
   chaises
</T>
<W id="w1" tokens="t1"
   pos="commonNoun"
   lemma="chaise"
   form="chaises"
   mstag="nP gF"/>
```

Note that all parts of speech are taken from the ISO registry [6] (Francopoulo et al., 2008). As in MAF, a word may refer to several tokens in order to represent multi-word units like *"pomme de terre"*. Conversely, a unique token may be refered by two different words in order to represent results of split based spelling correction like when *"unetable"* is smartly separated into the words *"une"* and *"table"*. The same configuration is required to represent correctly agglutination in fused prepositions like the token *"au"* that may be rewritten into the sequence of two words *"à" "le"*. On the contrary of MAF, cross-reference in token-word links for discontiguous spans is not allowed for the sake of simplicity. Let us add that one of our requirement is to have PASSAGE annotations mappable onto the MAF model and not to map all MAF annotations onto PASSAGE model. A **G** element denotes a syntactic group or a constituent (see details in section 3). It may be recursive or non-recursive and has an identifier, a type, and a content made of word forms or groups, if recursive. All group type values are taken from the ISO registry. Here is an example :

```
<T id="t0" start="0" end="3">
   Les
</T>
<T id="t1" start="4" end="11">
   chaises
</T>
<G id="g0" type="GN">
   <W id="w0" tokens="t0"/>
   <W id="w1" tokens="t1"/>
</G>
```

A group may also hold optional attributes like syntactic tagsets of MSTAG type. The syntactic relations are represented with a standoff annotations which refer to groups and word forms. A relation is defined by an identifier, a type, a source, and a target (see details in section 3. All relation types, like "subject" or "direct object" are mappable onto the ISO registry. An unrestricted number of comments may be added to any element by means of the mark element (i.e. M). Finally, a "Sentence"

[6] Data Category Registry, see http://syntax.inist.fr

element gathers tokens, word forms, groups, relations and marks and all sentences are included inside a “Document” element.

3 PASSAGE Syntactic Annotation Specification

3.1 Introduction

The annotation formalism used in PASSAGE[7] is based on the EASY one(Vilnat et al., 2004) which whose first version was crafted in an experimental project PEAS (Gendner et al., 2003), with inspiration taken from the propositions of (Carroll et al., 2002). The definition has been completed with the input of all the actors involved in the EASY evaluation campaign (both parsers’ developers and corpus providers) and refined with the input of PASSAGE participants. This formalism aims at making possible the comparison of all kinds of syntactic annotation (shallow or deep parsing, complete or partial analysis), without giving any advantage to any particular approach. It has six kinds of syntactic “chunks”, we call constituents and 14 kinds of relations The annotation formalism allows the annotation of minimal, continuous and non recursive constituents, as well as the encoding of relations wich represent syntactic functions. These relations (all of them being binary, except for the ternary coordination) have sources and targets which may be either forms or constituents (grouping several forms). Note that the PASSAGE annotation formalism does not postulate any explicit lexical *head*.

3.2 Constituent annotations

For the PASSAGE campaigns, 6 kinds of constituents (syntactic “chunks”) have been considered and are illustrated in Table 3.2:

- the Noun Phrase (GN for *Groupe Nominal*) may be made of a noun preceded by a determiner and/or by an adjective with its own modifiers, a proper noun or a pronoun;

- the prepositional phrase (GP, for *groupe prépositionnel*) may be made of a preposition and the GN it introduces, a contracted determiner and preposition, followed by the introduced GN, a preposition followed by an adverb or a relative pronoun replacing a GP;

- the verb kernel (**NV** for *noyau verbal*) includes a verb, the clitic pronouns and possible particles attached to it. Verb kernels may have different forms: conjugated tense, present or past participle, or infinitive. When the conjugation produces compound forms, distinct NVs are identified;

- the adjective phrase (**GA** for *groupe adjectival*) contains an adjective when it is not placed before the noun, or past or present participles when they are used as adjectives;

- the adverb phrase (**GR** for *groupe adverbial*) contains an adverb;

- the verb phrase introduced by a preposition (**PV**) is a verb kernel with a verb not inflected (infinitive, present participle,...), introduced by a preposition. Some modifiers or adverbs may also be included in PVs.

GN	- la très grande porte[8] (*the very big door*); - Rouletabille - eux (*they*), qui (*who*)
GP	- de la chambre (*from the bedroom*), - du pavillon (*from the lodge*) - de là (*from there*), dont (*whose*)
NV	- j’entendais (*I heared*) - [on ne l’entendait][9] plus (*we could no more hear her*) - Jean [viendra] (*Jean will come*) - [désobéissant] à leurs parents (*disobeying their parents*), - [fermée] à clef (*key closed*) - Il [ne veut] pas [venir] (*He doesn’t want to come*), - [ils n’étaient] pas [fermés] (*they were not closed*),
GA	- les barreaux [intacts] (*the intact bars*) - la solution [retenue] fut... (*the chosen solution has been...*), - les enfants [désobéissants] (*the disobeying children*)
GR	- aussi (*also*) - vous n’auriez [pas] (*you would not*)
PV	- [pour aller] à Paris (*for going to Paris*), - de vraiment bouger (*to really move*)

Table 1: Constituent examples

[7] Annotation guide: http://www.limsi.fr/Recherche/CORVAL/PASSAGE/eval_1/2007_10_05PEAS_reference_annotations_v11.12.html

3.2.1 Syntactic Relation annotations

The dependencies establish all the links between the minimal constituents described above. All participants, corpus providers and campaign organizers agreed on a list of 14 kinds of dependencies listed below:

1. subject-verb (**SUJ_V**): may be inside the same NV as between *elle* and *était* in *elle était* (*she was*), or between a GN and a NV as between *mademoiselle* and *appelait* in *Mademoiselle appelait* (*Miss was calling*);

2. auxiliary-verb (**AUX_V**), between two NVs as between *a* and *construit* in: *on a construit une maison* (*we have built a house*);

3. direct object-verb (**COD_V**): the relation is annotated between a main verb (NV) and a noun phrase (GN), as between *construit* and *la première automobile* in: *on a construit la première automobile* (*we have built the first car*);

4. complement-verb (**CPL_V**): to link to the verb the complements expressed as GP or PV which may be adjuncts or indirect objects, as between *en quelle année* and *construit* in *en quelle année a-t on construit la première automobile* (*In which year did we build the first car*);

5. modifier-verb (**MOD_V**): concerns the constituants which certainly modify the verb, and are not mandatory, as adverbs or adjunct clauses, as between *profondément* or *quand la nuit tombe* and *dort* in *Jean dort profondément quand la nuit tombe* (*Jean deeply sleeps when the night falls*);

6. complementor (**COMP**): to link the introducer and the verb kernel of a subordinate clause, as between *qu'* and *viendra* in *Je pense qu'il viendra* (*I think that he will come*); it is also used to link a preposition and a noun phrase when they are not contiguous, preventing us to annotate them as GP;

7. attribute-subject/object (**ATB_SO**): between the attribute and the verb kernel, and precising that the attribute is relative to (a) the subject as between *grand* and *est* in *il est grand*), or (b) the object as between *étrange* and *trouve* in *il trouve cette explication étrange*;

8. modifier-noun (**MOD_N**): to link to the noun all the constituents which modify it, as the adjective, the genitive, the relative clause... This dependency is annotated between *unique* and *fenêtre* in *l'unique fenêtre* (*the unique window*) or between *de la chambre* and *la porte* in *la porte de la chambre* (*the bedroom door*);

9. modifier-adjective (**MOD_A**): to relate to the adjective the constituents which modify it, as between *très* et *belle* in *¡la très belle collection* (*the very impressive collection*) or between *de son fils* and *fière* in *elle est fière de son fils* (*she is proud of her son*);

10. modifier-adverb (**MOD_R**): the same kind of dependency than MOD_A for the adverbs, as between *très* and *gentiment* in *elle vient très gentiment* (*she comes very kindly*);

11. modifier-preposition (**MOD_P**): to relate to a preposition what modifies it, as between *peu* and *avant* in *elle vient peu avant lui* (*she comes just before him*);

12. coordination (**COORD**): to relate the coordinate and the coordinated elements, as between *Pierre*, *Paul* and *et* in *Pierre et Paul arrivent* (*Paul and Pierre are arriving*);

13. apposition (**APP**): to link the elements which are placed side by side, when they refer to the same object, as between *le député* and *Yves Tavernier* in *Le député Yves Tavernier ...* (*the Deputy Yves Tavernier...*);

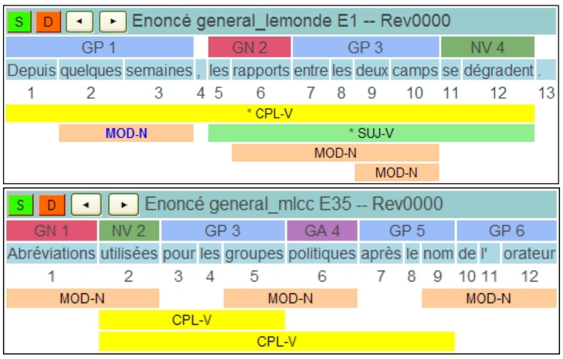
14. juxtaposition (**JUXT**): to link constituents which are neither coordinate nor in an apposition relation, as in enumeration. It also links clauses as *on ne l'entendait* et *elle était* in *on ne l' entendait plus ... elle était peut-être morte* (*we did not hear her any more... perhaps she was dead*).

Some dependencies are illustrated in the two annotated sentences illutrated in figure . These annotations have been made using EasyRef, a specific Web annotation tool developed by INRIA.

4 PASSAGE First Evaluation Campaign

4.1 Evalution Service

The first PASSAGE evaluation campaign was carried out in two steps. During the initial one-month development phase, a development corpus was used to improve the quality of



S D ◂ ▸ Enoncé general_lemonde E1 -- Rev0000

Depuis quelques semaines , les rapports entre les deux camps se dégradent .

S D ◂ ▸ Enoncé general_mlcc E35 -- Rev0000

Abréviations utilisées pour les groupes politiques après le nom de l' orateur

Figure 2: Example of two sentences annotations

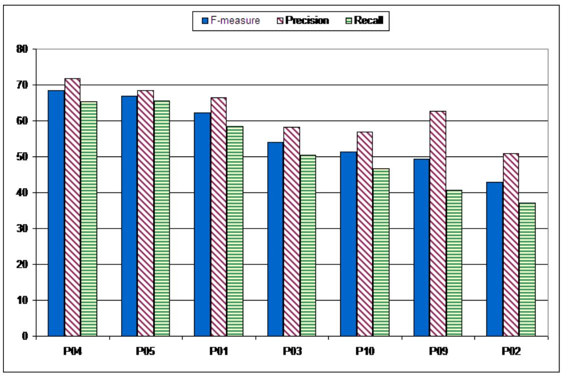


Figure 3: Overall functional relations results

parsers. This development corpus from the TECHNOLANGUE/EASY is composed of 40,000 sentences, out of which 4,000 sentences have been manually annotated for the gold standard. Based on these annotated sentences, an automatic WEB-based evaluation server provides fast performance feedback to the parsers' developers. At the end of this first phase, each participant indicated what he thought was his best parser run and got evaluated on a new set of 400 sentences selected from another part of the developement corpus which meanwhile had been manually annotated for the purpose and kept undisclosed.

The two phases represent a strong effort for the evaluators. To avoid adding the cost of managing the distribution and installation of the evaluation package at each developer's site, the solution of the WEB evaluation service was chosen. A few infrastructures have been already experimented in NLP, like GATE (Cunningham et al., 2002) infrastructures, but to our knowledge none has been used to provide an WEB-based evaluation service as PASSAGE did. The server was designed to manage two categories of users: parser developers and organizers. To the developers, it provides, almost in real time, confidential and secure access to the automatic evaluation of their submitted parses. To the organizers, it give access to statistics enabling them to follow the progress made by the developers, and easy management of the test phase. The evaluation server provides, through a simple WEB browser, access to both coarse and fine grain statistics to a developer's performance evaluation, globally for the whole corpus, at the level of a particular syntactic annotation or of a particular genre specific subcorpus, and also at the level of a single annotation for a particular word form.

4.2 Performance Results

Ten systems participated to the constituents annotation task. For most of the systems, F-measure is up to 90% and only three systems are between 80% and 90%. The trend is quite the same for Recall and Precision. Around 96.5% of the constituents returned by the best system are correct and it found 95.5% of the constituents present in gold standard. Figure 3 shows the results of the seven systems that participated to the functional relations annotation task. Performance is lower than for constituents and differences between systems are larger, an evidence that the task remains more difficult. No systems gets a performance above 70% in F-measure, three are above 60% and two above 50%. The last two systems are above 40%.

4.3 Systems Improvements

The higher system gets increasing results from the beginning of the development phase to the test phase for both constituents and relations. However, although the increase for relations is rather continuous, constituents results grow during the first few development evaluations, then reach a threshold from which results do not vary. This can be explained by the fact that the constituent scores are rather high, while for relations, scores are lower and starting from low scores.

Using the evaluation server, system improves its performance by 50% for the constituents and 600% for the relations, although performance vary according to the type of relation or constituent. Moreover, in repeating development evaluations, another consequence was the convergence of precision and recall.

5 Parser's outputs combination

The idea to combine the output of systems participating to an evalauation campaign in order to obtain a combination with better performance than the best one was invented to our knowledge by J. Fiscus (Fiscus, 1997) in a DARPA/NIST speech recognition evaluation (ROVER/Reduced Output Voting Error Reduction). By aligning the output of the participating speech transcription systems and by selecting the hypothesis which was proposed by the majority of the systems, he obtained better performances than these of the best system. The idea gained support in the speech processing community(Lööf et al., 2007) and in general better results are obtained with keeping only the output of the two or three best performing systems, in which case the relative improvement can go up to 20% with respect to the best performance (Schwenk and Gauvain, 2000). For text processing, the ROVER procedure was applied to POS tagging (Paroubek, 2000) and machine translation (Matusov et al., 2006).

In our case, we will use the text itself to realign the annotations provided by the various parser before computing their combination, as we did for our first experiments with the EASY evaluation campaign data (Paroubek et al., 2008). Since it is very likely taht the different parsers do not use the same word and sentence segmentation, we will realign all the data along a common word and sentence segmentation obtained by majority vote from the different outputs.

But our motivation for using such procedure is not only concerned with performance improvement but also with the obtention of a confidence measure for the annotation since if all systems agree on a particular annotation, then it is very likely to be true.

At this stage many options are open for the way we want to apply the ROVER algorithm, since we have both constituents and relations in our annotations. We could vary the selection order (between constituents and relations), or use different comparison functions for the sources/targets of constituents/relations(Patrick Paroubek, 2006), or perform incremental/global merging of the annoations, or explore different weightings/thresholding strategies etc. In passage, ROVER experiments are only beginning and we have yet to determine which is the best strategy before applying it to word and sentence free segmentation data. In the early experiment we did with the "EASy classic" PASSAGE track which uses a fixed word and sentence segmentation, we measured an improvement in precision for some specific subcorpora and annotations but improvement in recall was harder to get.

6 Conclusion

The definition of a common interchange syntactic annotation format is an essential element of any methodology aiming at the creation of large annotated corpora from the cooperation of parsing systems to acquire new linguistic knowledge. But the formalism aquires all of its value when backed-up by the deployment of a WEB-based evaluation service as the PASSAGE examples shows. 167 experiments were carried out during the development phase (around 17 experiments per participant in one month). The results of the test phase were available less than one hour after the end of the development phase. The service proved so successful that the participants asked after the evaluation, that the evaluation service be extended to support evaluation as a perennial service

References

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef Van Genabith, and Andy Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage pcfg-based lfg approximations. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 319–326, Barcelona, Spain, July.

Carroll, J., D. Lin, D. Prescher, and H. Uszkoreit. 2002. Proceedings of the workshop beyond parseval - toward improved evaluation measures for parsing systems. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175, Morristown, NJ, USA. Association for Computational Linguistics.

Declerck, T. 2006. Synaf: towards a standard for syntactic annotation. In *In proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May. ELRA.

Fiscus, Jonathan G. 1997. A post-processing system to yield reduced word error rates: recognizer output voting error reduction (rover). In *In proceedings of* *the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–357, Santa Barbara, CA.

Francopoulo, G., T. Declerck, V. Sornlertlamvanich, E. de la Clergerie, and M. Monachini. 2008. Data category registry: Morpho-syntactic and syntactic profiles. Marrakech. LREC.

Francopoulo, Gil. 2008. Tagparser: Well on the way to iso-tc37 conformance. In *In proceedings of the International Conference on Global Interoperability for Language Resources (ICGL)*, pages 82–88, Hong Kong, January.

Gendner, Véronique, Gabriel Illouz, Michèle Jardino, Laura Monceaux, Patrick Paroubek, Isabelle Robba, and Anne Vilnat. 2003. Peas the first instanciation of a comparative framework for evaluating parsers of french. In *Proceedings of the 10th Conference of the European Chapter fo the Association for Computational Linguistics*, pages 95–98, Budapest, Hungary, April. ACL. Companion Volume.

Hockenmaier, Julia and Mark Steedman. 2007. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.

Ide, N. and L. Romary. 2002. Standards for language ressources. Las Palmas. LREC.

Lööf, J., C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter, , and H. Ney. 2007. The rwth 2007 tc-star evaluation system for european english and spanish. In *In proceedings of the Interspeech Conference*, pages 2145–2148.

Matusov, Evgeny, N. Ueffing, and Herman Ney. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 158–165, Trento, Italy.

de la Clergerie, Eric, Christelle Ayache, Gaël de Chalendar, Gil Francopoulo, Claire Gardent, and Patrick Paroubek. 2008a. Large scale production of syntactic annotations for french. In *In proceedings of the First Workshop on Automated Syntactic Annotations for Interoperable Language Resources at IGCL'08*, pages 45–52, Hong Kong, January.

de la Clergerie, Eric, Olivier Hamon, Djamel Mostefa, Christelle Ayache, Patrick Paroubek, and Anne Vilnat. 2008b. Passage: from french parser evaluation to large sized treebank. In ELRA, editor, *In proceedings of the sixth international conference on Language Resources and Evaluation (LREC)*, Marrakech, Morroco, May. ELRA.

Miyao, Yusuke, Takashi Ninomiya, and Jun'ichi Tsujii. 2004. Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *In Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*.

Paroubek, Patrick, Isabelle Robba, Anne Vilnat, and Christelle Ayache. 2008. Easy, evaluation of parsers of french: what are the results? In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morroco.

Paroubek, Patrick. 2000. Language resources as by-product of evaluation: the multitag example. In *In proceedings of the Second International Conference on Language Resources and Evaluation (LREC2000)*, volume 1, pages 151–154.

Patrick Paroubek, Isabelle Robba, Anne Vilnat Christelle Ayache. 2006. Data, annotations and measures in easy - the evaluation campaign for parsers of french. In ELRA, editor, *In proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, pages 315–320, Genoa, Italy, May. ELRA.

Sagot, Benoît and Pierre Boullier. 2006. Efficient parsing of large corpora with a deep lfg parser. In *In proceedings of the sixth international conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, May. ELDA.

Schwenk, Holger and Jean-Luc Gauvain. 2000. Improved rover using language model information. In *In proceedings of the ISCA ITRW Workshop on Automatic Speech Recognition: Challenges for the new Millenium*, pages 47–52, Paris, September.

Vilnat, A., P. Paroubek, L. Monceaux, I. Robba, V. Gendner, G. Illouz, and M. Jardino. 2004. The ongoing evaluation campaign of syntactic parsing of french: Easy. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 2023–2026, Lisbonne, Portugal.

Constructing a Parser Evaluation Scheme

Laura Rimell and Stephen Clark
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford, OX1 3QD, United Kingdom
{laura.rimell,stephen.clark}@comlab.ox.ac.uk

Abstract

In this paper we examine the process of developing a relational parser evaluation scheme, identifying a number of decisions which must be made by the designer of such a scheme. Making the process more modular may help the parsing community converge on a single scheme. Examples from the shared task at the COLING parser evaluation workshop are used to highlight decisions made by various developers, and the impact these decisions have on any resulting scoring mechanism. We show that quite subtle distinctions, such as how many grammatical relations are used to encode a linguistic construction, can have a significant effect on the resulting scores.

1 Introduction

In this paper we examine the various decisions made by designers of parser evaluation schemes based on grammatical relations (Lin, 1995; Carroll et al., 1998). Following Carroll et al. (1998), we use the term *grammatical relations* to refer to syntactic dependencies between heads and dependents. We assume that grammatical relation schemes are currently the best method available for parser evaluation due to their relative independence of any particular parser or linguistic theory. There are several grammatical relation schemes currently available, for example Carroll et al. (1998), King et al. (2003), and de Marneffe et al. (2006). However, there has been little analysis of the decisions made by the designers in creating what turns out to be a complex set of dependencies for naturally occurring sentences. In particular, in this paper we consider how the process can be made more modular to help the parsing community converge on a single scheme.

The first decision to be made by the scheme designer is what types of linguistic constructions should be covered by the scheme. By *construction* we mean syntactic phenomena such as subject of verb, direct object of verb, passive voice, coordination, relative clause, apposition, and so on. In this paper we assume that the constructions of interest have already been identified (and there does appear to be broad agreement on this point across the existing schemes). A construction can be thought of as a unitary linguistic object, although it is often represented by several grammatical relations.

The second decision to be made is which words are involved in a particular construction. This is important because a subset of these words will be arguments of the grammatical relations representing the construction. Again, we assume that there is already broad agreement among the existing schemes regarding this question. One possible point of disagreement is whether to include empty elements in the representation, for example when a passive verb has no overt subject, but we will not address that issue here.

The next question, somewhat orthogonal to the previous one, and a source of disagreement between schemes, is how informative the representation should be. By *informative* we mean the amount of linguistic information represented in the scheme. As well as relations between heads, some schemes include one or more features, each of which expresses information about an individual head. These features can be the locus of richer linguistic information than is represented in the de-

Proceedings of the Coling 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 44–50
Manchester, August 2008

pendencies. A useful example here is tense and mood information for verbs. This is included in the PARC scheme, for example, but not in the Briscoe and Carroll or Stanford schemes; PARC is in general more informative and detailed than competing schemes. Although features are technically different from relations, they form part of an overall evaluation scheme and must be considered by the scheme designer. We will not consider here the question of how informative schemes should be; we only note the importance of this question for the resulting scoring mechanism.

The penultimate question, also a source of disagreement among existing schemes, is which words among all those involved in the construction should be used to represent it in the scheme. This decision may arise when identifying syntactic heads; for example, in the sentence *Brown said house prices will continue to fall*, we assume there is no disagreement about which words are involved in the clausal complement construction ({*said, house, prices, will, continue, to, fall*}), but there may be disagreement about which subset to use to represent the construction in the grammatical relations. Here, either *will* or *continue* could be used to represent the complement of *said*. This decision may also be theory-dependent to some degree, for example whether to use the determiner or the noun as the head of a noun phrase.

The final decision to make is the choice of relations and their arguments. This can also be thought of as the choice of how the set of representative words should be grouped into relations. For example, in a relative clause construction, the scheme designer must decide whether the relation between the relative pronoun and the head noun is important, or the relation between the relative pronoun and the verb, between the head noun and the verb, or some subset of these. The choice of label for each relation will be a natural part of this decision.

An important property of the representation, closely related to the choices made about representative words and how they are grouped into relations, is the number of relations used for a particular construction. We refer to this as the *compactness* property. Compactness essentially boils down to the valency of each relation and the information encoded in the label(s) used for the relation. We show that this property is closely related to the assigning of partial credit — awarding points even when a construction is not recovered completely correctly — and that it can have a significant effect on the resulting scoring mechanism.

The dividing lines between the various questions we have described are subtle, and in particular the last two questions (which words should represent the construction and which relations to use, and consequently how compactly the relations are represented) have significant overlap with one another. For example, if the auxiliary *are* in the passive construction *prices are affected* is chosen as one of the representative words, then a relation type which relates *are* to either *prices* or *affected* must also be chosen. For the relative clause construction *woman who likes apples and pears*, if the words and relations chosen include a representation along the lines of *relative-clause-subject(likes, woman)* and *subject(likes, who)*, then it is unlikely that the more compact relation *relative-clause(likes, woman, who)* would also be chosen. Despite the overlap, each question can provide a useful perspective for the designer of an evaluation scheme.

Decisions must be made not only about the representations of the individual constructions, but also about the interfaces between constructions. For example, in the sentence *Mary likes apples and pears*, the coordination structure *apples and pears* serves as direct object of *likes*, and it must be determined which word(s) are used to represent the coordination in the direct object relation.

We will illustrate some of the consequences of the decisions described here with detailed examples of three construction types. We focus on passive, coordination, and relative clause constructions, as analysed in the PARC (King et al., 2003), GR (Briscoe and Carroll, 2006), and Stanford (de Marneffe et al., 2006) evaluation schemes, using sentences from the shared task of the COLING 2008 parser evaluation workshop.[1] These three constructions were chosen because we believe they provide particularly good illustrations of the various decisions and their consequences for scoring. Furthermore, they are constructions whose representation differs across at least two of the three grammatical relation schemes under dicsussion, which makes them more interesting as examples. We believe that the principles involved, however,

[1] The shared task includes a number of additional formats besides the three grammatical relation schemes that we consider here, but the representations are sufficiently different that we don't consider a comparison fruitful for the present discussion.

apply to any linguistic construction.

We also wish to point out that at this stage we are not recommending any particular scheme or any answers to the questions we raise, but only suggesting ways to clarify the decision points. Nor do we intend to imply that the ideal representation of any linguistic construction, for any particular purpose, is one of the representations in an existing scheme; we merely use the existing schemes as concrete and familiar illustrations of the issues involved.

2 The Passive Construction

The following is an extract from Sentence 9 of the shared task:

> how many things are made out of eggs

We expect general agreement that this is a passive construction, and that it should be included in the evaluation scheme.[2] We also expect agreement that all the words in this extract are involved in the construction.

Potential disagreements arise when we consider which words should represent the construction. *Things*, as the head of the noun phrase which is the underlying object of the passive, and *made*, as the main verb, seem uncontroversial. We discard *how* and *many* as modifiers of *things*, and the prepositional phrase *out of eggs* as a modifier of *made*; again we consider these decisions to be straightforward. More controversial is whether to include the auxiliary verb *are*. PARC, for example, does not include it in the scheme at all, considering it an inherent part of the passive construction. Even if the auxiliary verb is included in the overall scheme, it is debatable whether this word should be considered part of the passive construction or part of a separate verb-auxiliary construction. Stanford, for example, uses the label auxpass for the relation between *made* and *are*, indicating that it is part of the passive construction.

The next decision to be made is what relations to use. We consider it uncontroversial to include a relation between *things* and *made*, which will be some kind of subject relation. We also want to represent the fact that *made* is in the passive voice, since this is an essential part of the construction and makes it possible to derive the underlying object position of *things*. If the auxiliary *are* is included, then there should be a verb-auxiliary relation between *made* and *are*, and perhaps a subject relation between *are* and *things* (although none of the schemes under consideration use the latter relation). PARC includes a variety of additional information about the selected words in the construction, including person and number information for the nouns, as well as tense and mood for the verbs. Since this is not included in the other two schemes, we ignore it here.

The relevant relations from the three schemes under consideration are shown below.[3]

PARC
passive(make, +)
subj(make, thing)

GR
(ncsubj made things obj)
(passive made)
(aux made are)

Stanford
nsubjpass(made, things)
auxpass(made, are)

PARC encodes the grammatical relations less compactly, with one subject relation joining *make* and *thing*, and a separate relation expressing the fact that *make* is in the passive voice. Stanford is more compact, with a single relation nsubjpass that expresses both verb-subject (via the arguments) and passive voice (via the label). GR has an equally compact relation since the obj marker signifies passive when found in the ncsubj relation. GR, however, also includes an additional feature passive, which redundantly encodes the fact that *made* is in passive voice.[4]

Table 1 shows how different kinds of parsing errors are scored in the three schemes. First note the differences in the "everything correct" row, which shows how many points are available for the construction. A parser that is good at identifying passives will earn more points in GR than in PARC and Stanford. Of course, it is always possible to look at accuracy figures by dependency type in order to understand what a parser is good at, as recommended by Briscoe and Carroll (2006), but it is

[2] PARC recognises it as an interrogative as well as a passive construction.

[3] Schemes typically include indices on the words for identification, but we omit these from the examples unless required for disambiguation. Note also that PARC uses the lemma rather than the inflected form for the head words.

[4] Although passive is technically a feature and not a relation, as long as it is included in the evaluation the effect will be of double scoring.

	PARC	GR	Stanf
Everything correct	2	3	2
Misidentify subject	1	2	1
Misidentify verb	0	0	0
Miss passive constr	1	1	0
Miss auxiliary	2	2	1

Table 1: Scores for passive construction.

also desirable to have a single score reflecting the overall accuracy of a parser, which means that the construction's overall contribution to the score is relevant.[5]

Observe also that partial credit is assigned differently in the three schemes. If the parser recognises the subject of *made* but misses the fact that the construction is a passive, for example, it will earn one out of two possible points in PARC, one out of three in GR (if it recognizes the auxiliary), but zero out of two in Stanford. This type of error may seem unlikely, yet examples are readily available. In related work we have evaluated the C&C parser of Clark and Curran (2007) on the BioInfer corpus of biomedical abstracts (Pyysalo et al., 2007), which includes the following sentence:

> Acanthamoeba profilin was cross-linked to actin via a zero-length isopeptide bond using carbodiimide.

The parser correctly identifies *profilin* as the subject of *cross-linked*, yet because it misidentifies *cross-linked* as an adjectival rather than verbal predicate, it misses the passive construction.

Finally, note an asymmetry in the partial credit scoring: a parser that misidentifies the subject (e.g. by selecting the wrong head), but basically gets the construction correct, will receive partial credit in all three schemes; misidentifying the verb, however (again, this would likely occur by selecting the wrong head within the verb phrase) will cause the parser to lose all points for the construction.

3 The Coordination Construction

The coordination construction is particularly interesting with regard to the questions at hand, both because there are many options for representing the construction itself and because the interface with other constructions is non-trivial. Here we consider an extract from Sentence 1 of the shared task:

> electronic, computer and building products

The coordination here is of nominal modifiers, which means that there is a decision to make about how the coordination interfaces with the modified noun. All the conjuncts could interact with the noun, or there could be a single relationship, usually represented as a relationship between the conjunction *and* and the noun.

Again we consider the decisions about whether to represent coordination constructions in an evaluation scheme, and about which words are involved in the construction, to be generally agreed upon. The choice of words to represent the construction in the grammatical relations is quite straightforward: we need all three conjuncts, *electronic*, *computer*, and *building*, and also the conjunction itself since this is contentful. It also seems reasonably uncontroversial to discard the comma (although we know of at least one parser that outputs relations involving the comma, the C&C parser).

The most difficult decision here is whether the conjuncts should be related to one another or to the conjunction (or both). Shown below is how the three schemes represent the coordination, considering also the interface of the coordination and the nominal modification construction.

PARC
adjunct(product, coord)
adjunct_type(coord, nominal)
conj(coord, building)
conj(coord, computer)
conj(coord, electronic)
coord_form(coord, and)
coord_level(coord, AP)

GR
(conj and electronic)
(conj and computer)
(conj and building)
(ncmod _ products and)

Stanford
conj_and(electronic, computer)
conj_and(electronic, building)
amod(products, electronic)
amod(products, computer)
amod(products, building)

[5] We assume that the overall score will be an F-score over all dependencies/features in the relevant test set.

Table 2 shows the range of scores assigned for correct and partially correct parses across the three schemes. A parser that analyses the entire construction correctly will earn anywhere from four points in GR, to seven points in PARC. Therefore, a parser that does very well (or poorly) at coordination will earn (or lose) points disproportionately in the different schemes.

	Parc	GR	Stanf
Everything correct	7	4	5
Misidentify conjunction	6	0	3
Misidentify one conjunct	6[a]	3	3[b]
Misidentify two conjuncts	5[a]	2	1

[a] The parser might also be incorrect about the coord_level relation if the conjuncts are misidentified.
[b] The score would be 2 if it is the first conjunct that is misidentified.

Table 2: Scores for coordination, including interface with nominal modification.

A parser that recognises the conjuncts correctly but misidentifies the conjunction would lose only one point in PARC, where the conjunction is separated out into a single coord_form relation, but would lose all four available points in GR, because the word *and* itself takes part in all four GR dependencies. Only two points are lost in Stanford (and it is worth noting that there is also an "uncollapsed" variant of the Stanford scheme in which the coordination type is not rolled into the dependency label, in which case only one point would be lost).

Note also an oddity in Stanford which means that if the first conjunct is missed, all the dependencies are compromised, because the first conjunct enters into relations with all the others. The more conjuncts there are in the construction, the more points are lost for a single parsing error, which can easily result from an error in head selection.

Another issue is how the conjuncts are represented relative to the nominal modifier construction. In PARC and GR, the conjunct *and* stands in for all the conjuncts in the modifier relation. This means that if a conjunct is missed, no extra points are lost on the modifier relation; whereas in Stanford, points are lost doubly – on the relations involving both conjunction and modification.

4 The Relative Clause Construction

For the relative clause construction, as for coordination, the choice of words used to represent the construction is straightforward, but the choice of relations is less so. Consider the following relative clause construction from Sentence 2 of the shared task:

> not all those who wrote

All three schemes under consideration use the set $\{$*those, who, wrote*$\}$ to describe this construction.[6]

PARC
pron_form(pro$_3$, those)
adjunct(pro$_3$, write)
adjunct_type(write, relative)
pron_form(pro$_4$, who)
pron_type(pro$_4$, relative)
pron_rel(write, pro$_4$)
topic_rel(write, pro$_4$)

GR
(cmod who those wrote)
(ncsubj wrote those _)

Stanford
nsubj(wrote, those)
rel(wrote, who)
rcmod(those, wrote)

Note that PARC represents the pronouns *who* and *those*, as it does all pronouns, at a more abstract level than GR or Stanford, creating a representation that is less compact than the others. GR and Stanford differ in terms of compactness as well: GR's cmod relation contains all three words; in fact, the ncsubj relationship might be considered redundant from the point of view of an evaluation scheme, since an error in ncsubj entails an error in cmod. Stanford's representation is less compact, containing only binary relations, although there is also a redundancy between nsubj and rcmod since the two relations are mirror images of each other.

For the sake of comparison, we include here two additional hypothetical schemes which have different characteristics from those of the three target schemes. In Hypothetical Scheme 1 (HS1), there are three relations: one between the head noun and the relative clause verb, one between the

[6] PARC also encodes the fact that pro$_3$ is a demonstrative pronoun, but we don't consider this part of the relative clause construction.

	PARC	GR	Stanf	HS1	HS2
Everything correct	7	2	3	3	1
Misidentify head noun	6	0	1	1	0
Misidentify verb	3	0	0	2	0
Miss relative clause construction	3	0	0	1	0

Table 3: Scores for relative clauses.

relative pronoun and the relative clause verb, and a third which relates the relative pronoun to the head noun. This third relation is not included in any of the other schemes. Hypothetical Scheme 2 (HS2) involves only one relation, which includes the same words as GR's cmod relation; the representation as a whole is quite compact since only one dependency is involved and it includes all three words.

Hypothetical Scheme 1
relative-subject(wrote, those)
subject(wrote, who)
relative-pronoun(those, who)

Hypothetical Scheme 2
relative-clause(wrote, those, who)

Table 3 shows the range of scores that can be attained in the different schemes. The total possible score varies from one for HS2, to three for Stanford and HS1, and up to seven for PARC.

Observe that any of the three types of error in Table 3 will immediately lose all points in both GR and HS2. Since all the schemes use the same set of words, this is due solely to the choice of relations and the compactness of the representations. Neither GR nor HS2 allow for partial credit, even when the parser assigns an essentially correct relative clause structure. This is a scenario which could easily occur due to a head selection error. For example, consider the following phrase from the shared task GENIA (Kim et al., 2003) data set , Sentence 8:

> . . . the RelA (p65) subunit of NF-kappa B , which activates transcription of the c-rel gene . . .

The C&C parser correctly identifies the relative clause structure, including the pronoun *which* and the verb *activates*, but incorrectly identifies the head noun as *B* instead of *subunit*.

Even between GR and HS2, which share the characteristic of not allowing for partial credit, there is a difference in scoring. Because GR starts with two dependencies, there is a loss of two points, rather than just one, for any error, which means errors in relative clauses are weighted more heavily in GR than in HS2.

Stanford also has a problematic redundancy, since the nsubj and rcmod relations are mirror images of each other. It therefore duplicates the GR characteristic of penalising the parser by at least two points if either the head noun or the relative clause verb is misidentified (in fact three points for the verb).

Observe also the asymmetry between misidentifying the head noun (one out of seven points lost in PARC, two out of three lost in Stanford and HS1) compared to misidentifying the verb (three points lost in PARC, all three lost in Stanford, but only one point lost in HS1). This reflects a difference between the schemes in whether the relative pronoun enters into a relation with the subject, the verb, or both.

5 Conclusion

In this paper we have shown how the design process for a relational parser evaluation scheme can be broken up into a number of decisions, and how these decisions can significantly affect the scoring mechanism for the scheme. Although we have focused in detail on three construction types, we believe the decisions involved are relevant to any linguistic construction, although some decisions will be more difficult than others for certain constructions. A direct object construction, for example, will normally be represented by a single relation between a verbal head and a nominal head, and indeed this is so in all three schemes considered here. This does not mean that the representation is trivial, however. The choice of which heads will represent the construction is important. In addition, Stanford distinguishes objects of prepositions from objects of verbs, while PARC and GR collapse the two into a single relation. Although part of speech information can be used to distinguish the two, a parser which produces PARC- or GR-style output in this regard will lose points in Stanford without some additional processing.

We have made no judgements about which decisions are best in the evaluation scheme design process. There are no easy answers to the questions raised here, and it may be that different solutions will suit different evaluation situations. We leave these questions for the parsing community to decide. This process may be aided by an empirical study of how the decisions affect the scores given to various parsers. For example, it might be useful to know whether one parser could be made to score significantly higher than another simply by changing the way coordination is represented. We leave this for future work.

References

Briscoe, Ted and John Carroll. 2006. Evaluating the accuracy of an unlexicalized statistical parser on the PARC DepBank. In *Proceedings of the ACL-Coling '06 Main Conf. Poster Session*, pages 41–48, Sydney, Austrailia.

Carroll, John, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st LREC Conference*, pages 447–454, Granada, Spain.

Clark, Stephen and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

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

Kim, Jin-Dong, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. 2003. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182.

King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, Budapest, Hungary.

Lin, Dekang. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*, pages 1420–1425, Montreal, Canada.

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

'Deep' Grammatical Relations for Semantic Interpretation

Mark McConville and **Myroslava O. Dzikovska**
Institute for Communicating and Collaborative Systems
School of Informatics, University of Edinburgh
Informatics Forum, 10 Crichton Street, Edinburgh, EH8 9AB, Scotland
{Mark.McConville,M.Dzikovska}@ed.ac.uk

Abstract

In this paper, we evaluate five distinct systems of labelled grammatical dependency against the kind of input we require for semantic interpretation, in particular for the deep semantic interpreter underlying a tutorial dialogue system. We focus on the following linguistic phenomena: passive, control and raising, noun modifiers, and meaningful vs. non-meaningful prepositions. We conclude that no one system provides all the features that we require, although each such feature is contained within at least one of the competing systems.

1 Introduction

The aim of the work reported in this paper is to evaluate the extent to which proposed systems of grammatical relations (GRs) reflect the kinds of deep linguistic knowledge required for semantic interpretation, in particular for deriving semantic representations suitable for domain reasoning in dialogue systems.

Grammatical relations either produced by or extracted from the output of wide-coverage syntactic parsers are currently used as input to shallow semantic parsers, which identify semantic relations that exist between predicators (typically verbs) and their dependents (Gildea and Jurafsky, 2002; Erk and Padó, 2006). Predicate-argument structure identified in this way can then be used in tasks like information extraction (Surdeanu et al., 2003) and question answering (Kaisser and Webber, 2007).

However, wide-coverage stochastic parsers are only rarely used in dialogue systems. Traditionally, interpretation modules of dialogue systems utilise specialised parsers and semantic interpreters handcrafted to a small domain (Seneff, 1992; Chang et al., 2002), or wide coverage deep parsers (Allen et al., 2007; Jordan et al., 2006; Wolska and Kruijff-Korbayová, 2003; Callaway et al., 2007; Kay et al., 1994). Unlike in information retrieval and question answering tasks, the system often needs to be connected to a knowledge base which represents the state of the world, and must be able to convert user utterances into knowledge base queries. In addition to identifying predicate-argument relationships, such systems need to support a variety of tasks, for example resolution of pronouns and anaphors, and interpreting negation, quantification, tense and modality.

While deep parsers produce precise semantic representations appropriate for such reasoning, they suffer from robustness problems. Wide-coverage dependency parsers could potentially provide a more robust alternative, provided that their output is easy to convert into semantic representations for reasoning.

Section 2 introduces the kind of deep linguistic processing application which motivates our approach to grammatical relations. Section 3 defines some underlying principles behind the kind of 'deep' GR system we have in mind. The remainder of the paper discusses a number of linguistic phenomena in detail, and evaluates how well various systems of GR representation from the dependency parsing literature capture the kind of linguistic insights required for interface with reasoning — passive (section 4), raising and control (section 5), noun modification (section 6) and syntactic versus semantic prepositions (section 7).

Proceedings of the Coling 2008 Workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 51–58
Manchester, August 2008

2 Motivation

As an example application that requires deep parsing consider a tutorial dialogue system that interprets students' answers to factual questions (e.g. *Which bulbs will be lit in this circuit?*) as well as explanation questions (e.g. *Explain your reasoning!*). It has been argued previously (Wolska and Kruijff-Korbayová, 2004; Rosé et al., 2003) that tutorial dialogue systems require deep understanding of student explanations, which can have significantly more complex structure than database queries in the information-seeking domain. In our application, if a student is asked for an explanation, his or her input has to be passed through the domain knowledge base to verify its factual correctness, and a separate process verifies that all relations mentioned in the explanation are correct and relevant. For example, imagine that the student says the following:

(1) The bulbs in circuits 1 and 3 will be lit because they are in closed paths with the batteries.

Here, the system has to verify two things: (a) that the facts are correct (bulbs in circuits 1 and 3 will be lit, and each of those bulbs is in a closed path with a battery); and (b) that the reason is valid — being in a closed path with a battery is a necessary and sufficient condition for a bulb to be lit.

This task is particularly interesting because it combines characteristics of deep and shallow interpretation tasks. On the one hand, the fact-checking mechanism requires a connection to the database. Thus, both pronouns and definite noun phrases need to be resolved to the objects they represent in the knowledge base, and first-order logic formulas representing utterance content need to be checked against the system knowledge. This task is similar to natural language interfaces to databases, or knowledge acquisition interfaces that convert language into knowledge base statements (Yeh et al., 2005). On the other hand, with respect to reason checking, human tutors have indicated that they would accept an answer simply if a student produces the key concepts and relations between them, even if the answer is not strictly logically equivalent to the ideal answer (Dzikovska et al., 2008). Human tutors tend to be especially lenient if a student is asked a generic question, like *What is the definition of voltage?*, which does not refer to specific objects in the knowledge base. Thus, a simpler matching mechanism is used to check the reasons, making this task more similar to an information retrieval task requiring shallower processing, i.e. that the predicate-argument relations are retrieved correctly (though negation still remains important).

Thus, while a specific task is used to motivate our evaluation, the conclusions would be applicable to a variety of systems, including both deep and shallow semantic interpreters.

For the purposes of this evaluation, we discuss features of grammatical representation relevant to two subtasks critical for the system: (a) identifying predicate-argument structure; and (b) resolving anaphora.

The extraction of predicate-argument relations is a common requirement for both shallow and deep semantic tasks. For example, for the student input in example (1) we may expect something like:[1]

(2)
```
(LightBulb b1) (LightBulb b2)
(lit b1 true) (lit b2 true)
(Path P3) (closed P3 true)
(contains P3 b1) (Path P4)
(closed P4 true) (contains P4 b2)
```

Resolving anaphora, on the other hand, is particularly important for the kind of deep semantic processing used in dialogue systems. Implicit in the above representation is the fact that the definite noun phrase *the bulbs in circuits 1 and 3* was resolved to domain constants `b1` and `b3`, and indefinite references to paths were replaced by Skolem constants `P3` and `P4`. The reference resolution process requires detailed knowledge of noun phrase structure, including information about restrictive modification, and this is the second focus of our evaluation.

Ideally, we would like a dependency parser to produce grammatical relations that can be converted into such semantic representations with minimal effort, thus minimising the number of specific rules used to convert individual relations. We discuss the principles underlying such representations in more detail in the next section.

[1] We used a simplified representation of quantifiers that assumes no scope ambiguity and uses skolem constants to represent existential quantification. This is sufficient for our particular application. In general, a more sophisticated quantifier representation would be necessary, for example that proposed in Copestake et al. (2005) or Bos and Oka (2002), but we leave the relevant evaluation for future work.

3 Deep grammatical relations

We formulated **four** principles for deep grammatical relations representation.

Firstly, grammatical relations should, whenever possible, reflect relations between the predicators (i.e. content words as opposed to function words) in a sentence. In addition, the same relation should correspond to the same role assignment. For example, the deep GRs in passive constructions should be the same as those in the active equivalents (see section 4), and the analysis of a control verb construction like *John persuaded Mary to dance* should make it clear that there is a 'subject' GR from *dance* to *Mary* similar to that in the implied sentence *Mary danced* (see section 5).

Secondly, a GR should, whenever possible, appear only if there is a an explicit selectional restriction link between the words. For example, in a raising verb construction like *John expects Mary to dance*, there should be **no** GR from the raising verb *expects* to its object *Mary* (see section 5). Also, where a preposition functions strictly as a syntactic role marker, as in the construction *John relies **on** Mary*, it should have no place in the GR analysis; rather there should be a direct link from the verb to the embedded noun phrase (see section 7).

Thirdly, the GRs should preserve evidence of syntactic modification to enable reference resolution. To understand why this is important, take the following two examples:

(3) The lit bulb is in a closed path.
The bulb in a closed path is lit.

From a pure predicate-argument structure perspective, these two sentences share exactly the same deep GRs:[2]

(4) `ext(lit,bulb)`
`ext(in-closed-path,bulb)`

However, from the perspective of reference resolution, the two sentences are very different. For the first example, this process involves first finding the lit bulb and then verifying that it is in a closed path, whereas for the second we need to find the bulb in a closed path and verify that it is lit. This difference can be captured by assigning the following additional deep GRs to the first example:

(5) `det(bulb,the)`
`ncmod(bulb,lit)`

And the following GRs are added to the analysis of the second example:

(6) `det(bulb,the)`
`ncmod(bulb,in-closed-path)`

Now the two analyses are formally distinct: (a) the first is rooted at predicate *in a closed path* and the second at *lit*; and (b) the definite external argument *the bulb* takes scope over the modifier *lit* in the first but over *in a closed path* in the second. Noun modification is discussed in section 6.

Finally, the set of grammatical relations should make it easy to identify and separate out constructions which are largely dependent on semantic/world knowledge, such as N-N modification, so that separate models and evaluations can be conducted as necessary.

4 Passive

The shared task dataset contains numerous passive participles, most of which can be classified into the following four groups depending on how the participle is used: (a) complement of passive auxiliary e.g. *Tax induction is **activated** by the RelA subunit*; (b) complement of raising verb e.g. *The administration doesn't seem **moved** by the arguments*; (c) nominal postmodifier e.g. *the genes involved in T-cell growth*; and (d) nominal premodifier e.g. *the proposed rules*.

In all these cases, our system for deep grammatical relation annotation requires: (a) that there is a relation **from** the passive participle **to** the deep object; and (b) that this relation be the same as in the corresponding active declarative construction, so that predicate-argument structure can be straightforwardly derived. Thus, for example, the analysis of *Tax induction is **activated** by the RelA subunit* will contain the GR `dobj(activated,induction)`, and that of *the proposed rules* will include `dobj(proposed,rules)`, where `dobj` is the relation between a transitive verb and its (deep) direct object.

We evaluated five GR-based output formats according to these two features. The results are presented in Table 1, where for each representation format (the rows) and each usage class of passive participles (the columns), we provide the GR which goes **from** the participle **to** its deep object,

[2] The representation is simplified for reasons of exposition. The GRs should be interpreted as follows: `ext` denotes the external argument of an adjective or preposition, `ncmod` a non-clausal restrictive modifier, and `det` the determiner of a noun.

	complement of passive auxiliary	complement of raising verb	nominal postmodifier	nominal premodifier	active
HPSG	ARG2 (of verb_arg12)				
RASP	ncsubj:obj				dobj
CCGBank	Spss\NP			N/N	S\NP/[NP]
Stanford	nsubjpass		-		dobj
PARC	subj		-		obj

Table 1: Representation of deep objects in passive and active

if such a GR exists.[3] The five GR representations compared are:

HPSG predicate-argument structures extracted from the University of Tokyo HPSG Treebank (Miyao, 2006)

RASP grammatical relations as output by the RASP parser (Briscoe et al., 2006)

CCGBank predicate-argument dependencies extracted from CCGBank (Hockenmaier and Steedman, 2007)

Stanford grammatical relations output by the Stanford Parser (de Marneffe et al., 2006)

PARC dependency structures used in the annotation of DepBank (King et al., 2003)

The first four columns in Table 1 represent, for each of the four uses of passive participles listed above, the grammatical relation, if any, which typically joins a passive participle to its deep object. The rightmost column presents the label used for this relation in equivalent active clauses. Adjacent columns have been collapsed where the same GR is used for both uses. The ideal system would have the same GR listed in each of the five columns.

The grammatical relations used in the Stanford, PARC and RASP systems are atomic labels like subj, obj etc, although the latter system does allow for a limited range of composite GRs like ncsubj:obj (a non-clausal surface subject which realises a deep object). In the HPSG system, verbal subjects and objects are represented as ARG1 and ARG2 respectively of strict transitive verb type verb_arg12. Finally, the GRs assumed in CCGBank consist of a lexical category (e.g. the strict transitive verb category S\NP/NP) with one argument emphasised. I assume the following notational convenience for those categories which contain specify more than one argument — the emphasised argument is surrounded by square brackets. Thus, subject and object of a strict transitive verb are denoted S\[NP]/NP and S\NP/[NP] respectively.

With respect to Table 1, note that: (a) in the CCGbank dependency representation, although prenominal passive participles *are* linked to their deep object (i.e. the modified noun), this relation is just one of generic noun premodification (i.e. N/N) and is thus irrelevant to the kind of predicate-argument relation we are interested in; (b) in the PARC and Stanford dependency representations, there is no GR from noun-modifying passive participles to their deep objects, just generic modification relations in the opposite direction; and (c) in PARC, passive participles are themselves marked as being passive, thus allowing a subsequent interpretation module to normalise the deep grammatical relations if desired.

If we are interested in a system of deep grammatical role annotation which allows for the representation of normalised GRs for passive participles in all their uses, then the HPSG Treebank format is more appropriate than the other schemes, since it uniformly uses deep GRs for both active and passive verb constructions. The RASP representation comes a close second, only requiring a small amount of postprocessing to convert ncsubj:obj relations into dobj ones. In addition, both the CCGBank and the Stanford notation distinguish two kinds of surface subject — those which realise deep subjects, and those which realise passivised deep objects.

5 Control

The shared task dataset contains a number of infinitives or participles which are dependents of non-auxiliary verbs or adjectives (rather than being noun modifiers for example). Most of these can

[3] The relations presented for HPSG and CCG are those for passive participle of *strict* transitive verbs.

	complements	adjuncts	raising
HPSG	✓	✓	×
RASP	✓	✓	×
CCGbank	✓	✓	×
Stanford	✓	×	✓
PARC	×	×	×

Table 2: Representation of controlled subjects and raising

be partitioned into the following three classes: (a) complements of subject control verbs e.g. *The accumulation of nuclear c-Rel* ***acts*** *to* ***inhibit*** *its own continued production*; (b) complements of subject raising verbs e.g. *The administration* ***seems moved*** *by arguments that . . .* ; and (c) subject controlled adjuncts e.g. *Alex de Castro has stopped by to* ***slip*** *six cards to the Great Man Himself.*

In all these cases, our deep grammatical role annotation requires that there be a subject relation (or an object relation in the case of a passive participle) from the infinitive/participle to the surface subject (or surface object in the case of object control) of the controlling verb/adjective. For example, the analysis of *Tax acts indirectly by inducing the action of various host transcription factors* will contain both the GRs `sbj(acts,Tax)` and `sbj(inducing,Tax)`. In addition, we also want to distinguish 'raising' verbs and adjectives from control structures. Thus, in the analysis of *The administration* ***seems moved*** *by arguments that . . .*, we want a (deep) object relation from *moved* to *administration*, but we don't want **any** relation from *seems* to *administration*.

We again evaluated the various GR-based output formats according to these features. The results are presented in Table 2, where for each representation format (the rows) we determine: (a) whether a verb with an understood subject which is a *complement* of the matrix verb is linked directly to its relevant subject (column 1); (b) whether a verb with an understood subject which is a controlled *adjunct* of the matrix verb is linked directly to its relevant subject (column 2); and (c) whether raising verbs are non-linked to their surface subjects (column 3). Note that the Stanford dependency representation is the only format which distinguishes between raising and control. This distinction is made both structurally and in terms of the name assigned to the relevant dependent — controlled subjects are distinguished from all other subjects (including raised ones) by having the label `xsubj` rather than just `nsubj`.[4]

The ideal GR representation format would have a tick in each of the three columns in Table 2. It is clear that no single representation covers all of our desiderata for a deep grammatical relation treatment of control/raising, but each feature we require is provided by at least one format.

6 Nominal modifiers

The dataset contains numerous prenominal modifiers[5], subdivided into the following three groups: (a) attributive adjectives e.g. *a few* ***notable*** *exceptions*; (b) verb participles e.g. *the* ***proposed*** *rules*; and (c) nouns e.g. *a* ***car*** *salesman*.

In order to ensure an adequate representation of basic predicate-argument structure, our system of deep grammatical annotation first of all requires that, from each prenominal adjective or verb, there is an appropriate relation **to** the modified noun, of the same type as in the corresponding predicative usage. For example, assuming that *He proposed the rules* has a direct object relation from *proposed* to *rules*, the same relation should occur in the analysis of *the proposed rules.* Similarly, if *The exceptions are notable* is analysed as having an external argument relation from *notable* to *exceptions*, then the same should happen in the case of *a few notable exceptions*. However, this does not appear to hold for prenominal nouns, since the relation between the two is not simply one of predication — a car salesman is not a salesman who 'is' a car, but rather a salesman who is 'associated' with cars in some way. Thus we would not want the same relation to be used here.[6]

Secondly, in order to ensure a straightforward interface with reference resolution, we need a modification relation going in the opposite direc-

[4] We have judged that CCGBank does not make the relevant distinction between raising and control verbs based on the dependency representations contained in the shared task dataset. For example, for the example sentence *The administration seem moved by the fact that . . .*, a CCG subject relation **is** specified from the raising verb *seem* to its surface subject *administration*.

[5] We focus on prenominal modifiers in order to keep the exposition simple. Similar remarks are valid for postnominal restrictive modifiers as well.

[6] Presumably the same goes for attributive adjectives which lack corresponding predicative uses, e.g. *the* ***former*** *president*.

tion, **from** the modified noun **to** each (restrictive) modifier, as argued in section 2. Thus, a complete GR representation of a noun phrase like *notable exceptions* would be cyclical, for example:

(7) `ext(notable,exceptions)`
`ncmod(exceptions,notable)`

We evaluated the various GR-based output formats according to these desiderata. The results are presented in Table 3. For each annotation scheme (the rows), we first present the relation (if any) which goes **from** the modified noun **to** each kind of premodifier (adjective, verb participle and noun respectively).[7] The middle three columns contain the relation (if any) which goes **to** the noun **from** each kind of modifier. Finally, the last three columns give the corresponding predicative relation used in the annotation scheme, for example in constructions like *The exceptions are notable*, *He proposed the rules*, or *Herbie is a car*. Where it is unclear whether a particular format encodes the relation between a predicative noun and its subject, we mark this as '?' in the last column.

Ideally, what we want is a representation where: (a) there is a GR in all nine columns (with the possible exception of the 'noun modifier to noun' one (column 6)); (b) the corresponding relations in the middle and righthand sections are identical, *except* for 'noun modifier to noun' (column 6) and 'predicative noun' (the last column) which should be distinct, since the relation between a noun modifier and its head noun is not simply one of predication.

It is clear that no one representation is perfect, though every feature we require is present in at least one representation system. Note in particular that the HPSG, PARC and Stanford systems are acyclic — the former only has 'modifier to noun' links, while the latter two only have 'noun to modifier' ones. The RASP format is cyclic, at least for prenominal participles — in *the proposed rules*, there is a modifier relation from *rules* to *proposed*, as well as a deep object relation from *proposed* to *rules*, the same relation that would be found in the corresponding predicative *the rules were proposed*.

Note finally that the PARC and Stanford representations distinguish between prenominal adjectives and nouns, in terms of the name of the relevant modifier GR. This corresponds well with our preference for a GR system where we can evaluate modules of N-N disambiguation (e.g. *luxury car salesman*) in isolation from other aspects of prenominal structure.

7 Prepositions

All five grammatical relations formats treat preposition phrases in pretty much the same way: (a) there is a GR link from the head of which the PP is a complement or modifier to the preposition itself (the HPSG representation has this link going in the opposite direction for PP modifiers, but the principle is the same); and (b) there is a link from the preposition to its complement NP. For example, the noun phrase *experts in Congress* is annotated as follows:

(8) `ncmod(experts,in)`
`dobj(in,Congress)`

The only PPs which have been handled differently are agentive *by*-PPs of passive participles, which are either normalised or treated using a special, construction-specific GR.

Note however that all prepositions are not equal when it comes down to representing the predicate-argument structure of a sentence. In a nutshell, some prepositions are predicators (e.g. *experts **in** Congress*) whereas others are simply syntactic role markers (e.g. *a workout **of** the Suns*). Ideally, we would want a GR system which marks this distinction, for example by annotating predicator prepositions as lexical heads and ignoring role-marking prepositions altogether. The only GR scheme which attempts to make this distinction is the PARC system, which has a `ptype` feature for every preposition with two possible values, `semantic` and `non-semantic`. However, this does not appear to have been annotated consistently in the PARC dataset — the only examples of non-semantic prepositions are agentive *by*-PPs of passive participles.

8 Conclusion

We have proposed a set of principles for developing a grammatical relation annotation system for use with both shallow and deep semantic interpretation systems, in particular a tutorial dialogue system. We then evaluated five different GR schemes from the dependency parsing literature based on how well they handle a number of 'deep' syntactic phenomena implied by these principles,

[7] Note that the `N/N` links in the CCG representation actually go from the modifier to the noun. However, they have been included in the set of 'noun to modifier' relations since they are formally modifier categories (i.e. of the form X/X).

	noun to modifier			modifier to noun			predicative		
	A	V	N	A	V	N	A	V	N
RASP	ncmod			-	ncsubj etc	-	-	ncsubj etc	-
HPSG	-			a_arg1	v_arg1 etc	n_arg1	a_arg1	v_arg1 etc	n_arg1
CCG	N/N	-	N/N	-	S\NP etc	-	Sadj\NP	S\NP etc	?
PARC	adjunct		mod	-			subj	subj	?
Stanf	amod		nn	-			nsubj	nsubj	?

Table 3: Representation of prenominal modifiers

i.e. passive, control and raising, noun modification, and meaningful vs. non-meaningful prepositions. We conclude that none of the proposed GR annotation schemes contains everything we require for deep semantic processing, although each of the features/distinctions we included in our list of desiderata is provided by at least one system.

Many of the deep syntactic phenomena discussed here are known issues for shallow semantic tasks like semantic role labelling. For example, passive constructions are a recognised source of noise in semantic role labelling systems (Gildea and Jurafsky, 2002), and resolving controlled subjects provides more data for training models of selectional restrictions, which are known to be useful features for role labelling. More generally, Chen and Rambow (2003) demonstrate that a focus on 'deep' syntactic features results in a more accurate stochastic semantic role labeller than using surface information alone.

Note also that the deep grammatical role representation proposed here is meant to be 'theory-neutral', in the sense that it was not influenced by any one of the competing grammar formalisms to the exclusion of the others. Indeed, it should be a straightforward task to write a grammar using either the HPSG, LFG, CCG or RASP-style underlying formalism which can produce an output representation consisting of deep relations, constructed in a purely compositional manner. Indeed, the syntactic phenomena discussed in this paper are those which form the basis of numerous introductory textbooks on English generative syntax (Haegeman, 1994; Sag and Wasow, 1999; Bresnan, 2000). In addition, the phenomena which form the basis of the analysis in this paper were among those which had been the focus of a significant amount of attention in the development of the semantic interpretation system underlying our domain-independent tutorial dialogue system. Other issues which were considered, but for which we lack space to discuss in detail include: (a) expletive pronouns should be ignored, i.e. the subject pronouns in 'impersonal' verb constructions like *It is raining* or *It's great that John loves Mary* should not be seen as the target of deep grammatical relations; (b) unbounded dependencies should be resolved, i.e. in the relative clause *the woman Bill thinks John loves* there should be an object relation between the embedded verb *loves* and its extracted object *woman*; (c) restrictive and non-restrictive modification (including apposition) should be distinguished, since the latter is not relevant for reference resolution; and (d) certain subsentential conjunctions need to be compiled out (for examples like *electronic, computer and building products*).

Finally, we recognise that, in many cases, it is possible to transform parser representations into our desired format. For example, if the parser output tells us that a given verb form is a passive participle, we can use this information to remap the surface relations, thus retrieving the underlying predicate-argument structure. However, we prefer a system where this kind of post-processing is not needed. Reasons for this include the increased potential for error in a system relying on post-processing rules, as well as the need to have both detailed documentation for how each parser output format handles particular constructions, as well as a comprehensive mapping schema between representations. Having a community standard for GR-based parser output is an essential element of future parsing technology, and to be practically useful in a range of semantic interpretation tasks, this standard should involve 'deep' syntactic distinctions of the kind discussed in this paper.

9 Acknowledgements

The work reported here was supported by grants N00014-08-1-0179 and N00014-08-1-0043 from the Office of Naval Research.

References

Allen, James, Myroslava Dzikovska, Mehdi Manshadi, and Mary Swift. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the ACL'07 Workshop on Deep Linguistic Processing*.

Bos, Johan and Tetsushi Oka. 2002. An inference-based approach to dialogue system design. In *Proceedings of COLING'02*.

Bresnan, Joan. 2000. *Lexical-Functional Syntax*. Basil Blackwell.

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL'06 Interactive Presentation Sessions*.

Callaway, Charles B., Myroslava Dzikovska, Elaine Farrow, Manuel Marques-Pita, Colin Matheson, and Johanna D. Moore. 2007. The Beetle and BeeDiff tutoring systems. In *Proceedings of SLaTE'07*.

Chang, N., J. Feldman, R. Porzel, and K. Sanders. 2002. Scaling cognitive linguistics: Formalisms for language understanding. In *Proceedings of ScaNaLU'02*.

Chen, John and Owen Rambow. 2003. Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proceedings of EMNLP'03*.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3:281–332.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*.

Dzikovska, Myroslava O., Gwendolyn E. Campbell, Charles B. Callaway, Natalie B. Steinhauser, Elaine Farrow, Johanna D. Moore, Leslie A. Butler, and Colin Matheson. 2008. Diagnosing natural language answers to support adaptive tutoring. In *Proceedings of FLAIRS'08 special track on Intelligent Tutoring Systems*.

Erk, Katrin and Sebastian Padó. 2006. SHALMANESER - a toolchain for shallow semantic parsing. In *Proceedings of LREC'06*.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3).

Haegeman, Liliane. 1994. *Introduction to Government and Binding Theory*. Basil Blackwell, 2nd edition edition.

Hockenmaier, Julia and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3).

Jordan, Pamela, Maxim Makatchev, Umarani Pappuswamy, Kurt VanLehn, and Patricia Albacete. 2006. A natural language tutorial dialogue system for physics. In *Proceedings of FLAIRS'06*.

Kaisser, Michael and Bonnie Webber. 2007. Question answering based on semantic roles. In *Proceedings of the ACL'07 Workshop on Deep Linguistic Processing*.

Kay, Martin, Jean Mark Gawron, and Peter Norvig. 1994. *Verbmobil: A Translation System for Face-To-Face Dialog*. CSLI Press, Stanford, CA.

King, Tracy Holloway, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 dependency bank. In *Proceedings of EACL'03*.

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

Rosé, C. P., D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. 2003. The role of why questions in effective human tutoring. In *Proceedings of AIED'03*.

Sag, Ivan A. and Thomas Wasow. 1999. *Syntactic Theory: A Formal Introduction*. CSLI.

Seneff, Stephanie. 1992. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1).

Surdeanu, Mihai, Sanda M. Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of ACL'03*.

Wolska, Magdalena and Ivana Kruijff-Korbayová. 2003. Issues in the interpretation of input in mathematical dialogs. In Duchier, Denys, editor, *Prospects and advances in the syntax/semantics interface. Lorraine-Saarland Workshop Series proceedings.*

Wolska, Magdalena and Ivana Kruijff-Korbayová. 2004. Analysis of mixed natural and symbolic language input in mathematical dialogs. In *Proceedings of ACL'04*.

Yeh, Peter Z., Bruce Porter, and Ken Barker. 2005. Matching utterances to rich knowledge structures to acquire a model of the speaker's goal. In *Proceedings of K-CAP'05*.

Author Index