WikiWoods: Syntacto-Semantic Annotation for English Wikipedia

Dan Flickinger[♣], Stephan Oepen[♠], Gisle Ytrestøl[♠]

[♣] Stanford University, Center for the Study of Language and Information
[♠] University of Oslo, Department of Informatics
danf@stanford.edu, oe@ifi.uio.no, gisley@ifi.uio.no

Abstract

WikiWoods is an ongoing initiative to provide rich syntacto-semantic annotations for English Wikipedia. We sketch an automated processing pipeline to extract relevant textual content from Wikipedia sources, segment documents into sentence-like units, parse and disambiguate using a broad-coverage precision grammar, and support the export of syntactic and semantic information in various formats. The full parsed corpus is accompanied by a subset of Wikipedia articles for which gold-standard annotations in the same format were produced manually. This subset was selected to represent a coherent domain, Wikipedia entries on the broad topic of Natural Language Processing.

1. Background—Motivation

Wikipedia is met with great interest by researchers in (computational) linguistics; it provides a massive and relatively high-quality collection of text and (predominantly unstructured) encyclopedic knowledge.[1] To facilitate research grounded in this community resource, we are working to provide automatically created 'deep' annotations for the full English Wikipedia. This information is obtained from a broad-coverage parsing system couched in the HPSG framework—the LinGO English Resource Grammar (ERG; Flickinger, 2000)—making available detailed syntactic analyses as well as basic propositional Minimal Recursion Semantics (MRS; Copestake, Flickinger, Pollard, & Sag, 2005). While textual versions of Wikipedia with various types of annotation are available from related initiatives already, WikiWoods transcends existing resources in three aspects:[2] (a) we consider the task of extracting relevant linguistic content from Wikipedia sources a relevant research question in its own right, aiming for comparatively high-quality text; (b) the WikiWoods syntacto-semantic annotations are considerably 'deeper' (richer in linguistic granularity and generalizing further over surface-level properties) than part-of-speech or syntactic dependency information; and (c) for a domain-specific subset of Wikipedia, we provide hand-corrected (i.e. gold-standard) sentence segmentation and annotations in the exact same format and depth.

Format of Annotations The type of annotations available in WikiWoods is exemplified by Figures 1 and 2. Internally, each full HPSG analysis is characterized by the derivation tree (left in Figure 1), labeled with identifiers of HPSG constructions (at interior nodes) and lexical entries (at leaf nodes). When combined with the grammar itself, the derivation provides an unambiguous 'recipe' for invoking and combining appropriately the rich linguistic constraints encoded by the ERG, a process that results in an HPSG typed feature structure with, on average, about 250 attribute – value pairs (including detailed accounts of morpho-syntactic properties, subcategorization information, other grammaticalized properties at the lexical and phrasal levels, and a compositional approach to propositional semantics). At the same time, we anticipate that just the abstract labels of the derivation provide valuable information by themselves, as they analyse syntactic structure in broad types of constructions, e.g. subject – head, specifier – head, head – complement, and adjunct – head in the top nodes of Figure 1.

A more conventional representation of syntactic information is available in the form of constituent trees labeled with 'classic' category symbols (right in Figure 1), using an inventory of 78 distinct labels in the default configuration. Conceptually, these labels abbreviate salient properties of the full HPSG feature structures, and there is technology support for customization of this process. In a nutshell, a technologically somewhat savvy user can adapt the templates used in mapping specific feature structure configurations to abbreviatory category symbols and re-run the labeling process, i.e. obtain a custom set of constituent trees from the original derivations. Such customization could also include transformations of tree structure, for example flattening VPs (which the ERG analyzes as binary branching) or removal of category-preserving unary projections.

In terms of semantic annotation available in WikiWoods, Figure 2 shows the (not yet scope-resolved) MRS logical form for the same sentence. Loosely speaking, there are three types of logical variables in this representation, events ('e_i'), instances ('x_j'), and handles ('h_k'). Of these, the latter serve a formalism-internal function, encoding scopal relations and facilitating underspecification (for formal details see Copestake et al., 2005), but will be ignored here—as are the specifics of quantifier representations (the '_q' relations in Figure 2). Events in MRS denote states or activities (and have spatio-temporal extent), while instance variables will typically correspond to entities. The latter types of variables typically carry (semantic reflexes of) morpho-

[1] In the past few years, the use of Wikipedia content for a variety of research tasks has seen a lively increase; http://tinyurl.com/mkbergman provides an overview of recent Wikipedia-based R&D, most from a Semantic Web point of view.

[2] The WikiWoods on-line pages (see below) provide links to related initiatives, and the authors would be delighted to receive additional pointers.

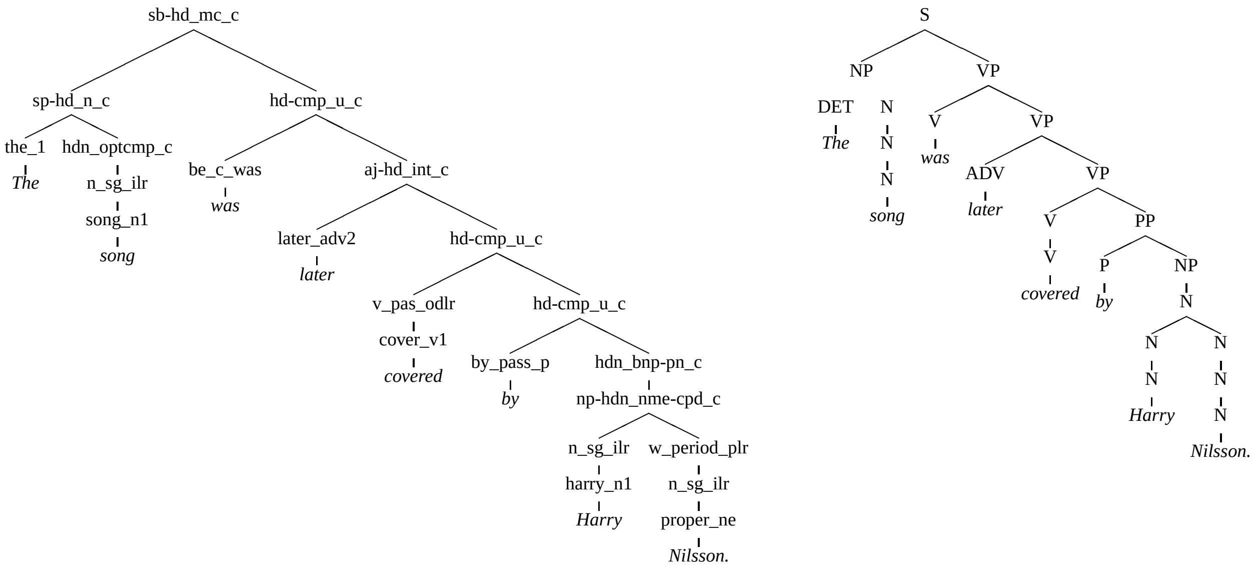

Figure 1: Syntactic representations for *The song was later covered by Harry Nilsson.* The HPSG derivation (left) is labeled with identifiers of lexical entries and constructions; the phrase structure tree (right) reduces HPSG signs to conventional category labels.

syntactic information: tense, mood, and aspect, or person and number, on events and instances, respectively. Reflecting meaning composition from words and phrases, the two-place *compound_name* relation provides the bracketing of the complex proper name; however, syntax does not necessarily determine the exact internal structure of complex nominals, e.g. the (likely) interpretation of *Harry* as a first name, in this case. Finally, observe how at the level of semantics the role assignments are normalized: the mapping of syntactic functions to semantic arguments is reversed in the passive construction, but at the MRS level the passive and active variants receive identical semantics—as would be the case with other diathesis alternations analyzed by the grammar, e.g. the dative shift in *Kim gave Sandy a book.* vs. *Kim gave a book to Sandy.* At the same time, word sense distinctions are rarely grammaticalized (for example the ‘spread over’ vs. performing arts senses of *cover*) and hence remain underspecified.

Article Outline Deep parsing technology has matured to a point where a good balance of grammatical coverage, parsing efficiency, and output accuracy can be obtained for large volumes of running text. Still, adapting the technology to the scale of Wikipedia presents a number of engineering challenges. In the following, we provide a high-level summary of our approach to preprocessing, parsing, and annotating Wikipedia (Section 2.), give a short overview of the hand-annotated subset (Section 3.), and then discuss in some detail our approach to scaling to the full Wikipedia, including some core statistics and a preliminary analysis of the quality of annotations available (Section 4.). As part of the concluding remarks, we speculate about remaining opportunities for improving and extending the WikiWoods resource (Section 5.).

Our complete toolchain and the gold-standard sub-corpus are already available as part of the DELPH-IN open source toolchain; we expect to publicly release the complete parsed Wikipedia, dubbed WikiWoods, in time for LREC 2010.[3]

2. Preprocessing Wikipedia

Natively, Wikipedia articles are edited in a markup language that combines a comparatively simple core with select elements from LaTeX and HTML. A template facility (see below for details) provides functionality similar to LaTeX macros (which, in the extreme, resemble a general programming language). However, the use of templates in general (and non-trivial ones in particular) appears comparatively limited in Wikipedia sources; this may be owed to the diverse authorship of Wikipedia, where the vast majority of contributors are non-experts in terms of the markup language.

In preparing the WikiWoods corpus, our abstract goal is to extract textual content, while suppressing irrelevant or strictly non-linguistic information. Naturally, it is difficult to operationalize our notions of ‘relevant’ or ‘non-linguistic’. As a general guiding principle, we try to: (a) reflect in the WikiWoods text corpus what is actually displayed to readers of Wikipedia, i.e. the content (and sequential ordering of content elements) rendered for display by the interplay of Wikipedia server software and the web browser; and to (b) preserve textual content elements (for example headings, paragraphs, or itemized lists) but remove content elements that have a predominantly idiosyncratic, non-linguistic structure (e.g. various kinds of summary boxes and other tabular data).

For preprocessing Wikipedia source files, we opted to operate predominantly at a textual level, i.e. working with pattern matching at the level of concrete syntax, rather than actually parsing all Wikipedia markup into abstract syntax (and then operating on parse trees). We discuss the benefits and inherent limitations of this design choice in Section 5.

[3] See `http://www.delph-in.net/wikiwoods/` for details.

below, but essentially it is a matter of convenience, robustness (to ill-formed markup), and processing efficiency. While we believe our solution arrives at quite high-quality text, there are certain subtle aspects of Wikipedia markup that we cannot handle properly in this setup. We will point out examples of such corner cases in the discussion below, together with estimates of the frequency and relative impact of the resulting (minor) deficiencies in our textual output.

Section and Markup Pruning In preparing the corpus we aim for a practical balance between (a) preserving potentially relevant markup and (b) presenting the data in a form that is easily accessible to both human readers and NLP tools. From the raw source files of Wikipedia articles, we eliminate markup that is linguistically irrelevant. This includes, for example, most meta information (category and inter-language links, say, or pointers to disputed facts and background discussion), in-text comments, tables, displayed code blocks, and inline image data. But we aim to preserve all markup that *could* be important for linguistic analysis. Markup corresponding to itemized lists, (sub)headings, hyperlinks, or specific font properties, for example, often signals specialized syntax; the use of italics, for example, often indicates foreign-language expressions or use – mention contrasts (see below for the interface of relevant markup and grammatical analysis).

Applying a rich cascade of regular expressions on each article, unwanted markup and in some cases entire sections from the source articles are removed in preprocessing. Templatic sections like *See Also*, *References* or *Bibliography*, and *External Links*, in our view, have no bearing on linguistic analysis. In fact, these sections typically occur towards the end of an article—in a sense delineating the article body—and preprocessing will therefore remove any remaining content elements following the above sections.

Tables While the bulk of Wikipedia content is unstructured text, there is a non-trivial amount of semi-structured data, which typically is presented in tabular form—for example so-called ‘info boxes’ which provide semi-standardized summaries for a given topic group, for example key biographical facts in articles about a specific person. Mining info boxes and other templatic content elements in Wikipedia can be a good way of extracting formal, ontological knowledge from Wikipedia. However, the focus of our WikiWoods initiative is on supporting NLP research and specifically methods that make use of syntactic or semantic analysis of unstructured text. Hence, all tabular content elements are removed in preprocessing. The mixing and matching of Wikipedia markup for tables and HTML table syntax (and interaction with the Wikipedia template facility, see below) make the detection of tabular content a non-trivial task. Furthermore, Wikipedia contributors (naturally) do not always observe the correct markup syntax, and it appears the Wikipedia server software (converting Wikipedia markup to HTML) and browser-side HTML rendering treat incorrect markup robustly in many cases. Table start and table end tags, for example, do not match up properly in thousands of cases, particularly so when tables are nested inside each other. To handle such source-level errors, we experimented with a number of heuristic recovery techniques, in a sense aiming to mimic part of the robustness of the Wikipedia server software and HTML browsing. We believe our final solution strikes a good balance of removing the by far vast majority of tabular content elements, without inadvertently removing non-tabular content. Table processing is part of our open-source preprossing package[4], and the specific heuristics are documented as part of the source code.

Templates Wikipedia templates provide a macro-like facility, aiming to simplify the presentation of schematic or repetitive content, and of course seeking to assist authors in overall consistency. Template processing is arguably the biggest remaining challenge to our decision of preprocessing Wikipedia source files through string-level pattern matching. Quite a large number of templates are ‘benign’, either not contributing visual content at all (hence, only hidden metadata), or packaging tabular and otherwise schematic content—that according to our interpretation of ‘linguistic relevance’ will be filtered out during preprocessing anyway. Another class of templates does not necessarily add content but serves to bracket sub-sequences of text with non-standard rules of interpretations, for example foreign-language expressions (which, when wrapped in a template announce the specific language and maybe script, are more likely to be rendered correctly in HTML), phonetic transcriptions (in IPA), or in-text blocks of program code. For these cases, we preserve the template ‘as is’ during preprocessing; our parsing setup employs a customized tokenizer for Wikipedia, which extracts the actual strings from these templates and interfaces bracketing information into the parser, where appropriate.[5]

Finally, there is a class of templates that cannot be handled correctly in our approach. The *convert* template, for example, can be used for various unit conversions (miles to kilometers, say), where a wealth of template parameters control display options, the application to invididual units of measure or ranges, etc. Likewise, a family of templates for dates of birth can calculate the current age (or a living person), given the current date—i.e. producing dynamic content. Expanding such cases correctly would require a complete implementation of the Wikipedia template mechanism, in other words large parts of the Wikipedia server software. Based on a review of (estimated) template frequency after preprocessing and an inspection of typical use patterns, we preserve some frequent instances of this last class ‘as is’, while removing all others. Unconditional removal of contentful templates, in principles, carries the risk of ‘damaging’ linguistic content, for example deleting a mandatory constituent in an otherwise wellformed utterance. Given the observed frequencies and our analysis of common context conditions on template usage, we believe that a very small proportion of utterances in the Wikiwoods corpus is affected, probably on the order of one tenth of a percent.

[4] See `http://www.delph-in.net/wikiwoods/` for download and installation instructions.

[5] Again, please see the WikiWoods on-line site for details on template handling, including a list over Wikipedia templates kept in the WikIWoods corpus, together with the set of finite-state rules used for preparing the actual input to the deep parser.

Textual Exchange Format While several Wikipedia markup parsers and format converters exist (typically mapping into XML), we opt for a compact exchange format in plain text (including Wikipedia markup preserved by preprocessing). This is in part to better balance human and machine readability, but more importantly because a rigid XML format would create formal obstacles. The central unit of analysis in our work is the sentence (or otherwise independent utterance), and the WikiWoods exchange format represents one sentence per line. Thus, we follow the Un*x philosophy: any sequence of utterances or concatenation thereof represents a well-formed structural unit.

After preprocessing has extracted 'core' linguistic content from an article, we break the text into utterances (i.e. sentence-like units) through the combination of an off-the-shelf sentence segmenter[6] and another layer of Wikipedia-specific regular expressions (the *tokenizer* tool, by default, expects 'pure' text without markup). For instance, the tool initially failed to insert segment boundaries between numbered list elements (where the Wikipedia markup starting new list items, in a sense, takes on the function of sentence-initial punctuation). Finally, in our textual exchange format, multi-whitespace sequences are normalized to single spaces, and each utterance is represented as one line of its own. See Ytrestøl, Flickinger, & Oepen (2009) for further technical detail on this stage of preprocessing.

Corpus Organization In part to improve filesystem performance, in part to logically divide the corpus into units of 'manageable' size, articles are grouped into *segments*, each comprised of 100 consecutive articles (after sorting article titles lexicographically). Articles are numbered using seven-digit identifiers, starting from 0000100, whereas segments are numbered using five-digit identifiers, starting from 00101 (the gold-standard WeScience articles and segments populate the identifier space below these starting values; see below). Article identifiers are constructed to reflect the corresponding segment identifier. Within each article, a globally unique identifier is assigned to each utterance, where utterance identifiers are internally structured so as to allow easy retrieval of the underlying article (and segmment). We allocate five digits for utterance identifiers, but number in increments of 10. This is a convention also used in the WeScience sub-corpus (see below), which leaves open the possibility of hand-correcting utterance segmentation without having to adjust identifiers globally.

We organize the WikiWoods corpus as a collection of text files, using the above conventions. The corpus is available in several formats: (a) *raw* article files, named by article identifier and Wikipedia article title; (b) textual exchange files (the result of preprocessing and segmentation), one per segment; and (c) Redwoods treebank directories (see below), one per segment.

3. WeScience: Facts and Figures

Ytrestøl et al. (2009) present the WeScience Treebank, a selection of 100 Wikipedia articles (in the broad domain of NLP) that were preprocessed in the exact manner described above and then paired with hand-corrected sentence segmentation and gold-standard HPSG analyses.[7] Manual annotation in WeScience applies the discriminant-based LinGO Redwoods treebanking tools (Oepen, Flickinger, Toutanova, & Manning, 2004; Carter, 1997). In Redwoods, the treebank records the complete syntacto-semantic analyses provided by the ERG, and there are tools to 'export' different kinds of linguistic representation (at variable granularity), including those of Figures 1 and 2. In a nutshell, annotation in Redwoods amounts to manual disambiguation, i.e. identifying the correct analysis among the forest of possible HPSG analyses. Consequently, out-of-scope inputs that the grammar cannot parse create 'gaps' in treebank coverage (but see Section 5. for future work addressing this challenge).

The WeScience Treebank can be viewed as a gold-standard sub-corpus of the full WikiWoods, where manual correction and disambiguation provide (a) higher-quality annotations, quite generally, (b) estimates of expected error rates, and (c) training data for statistical parse disambiguation. The 100 articles in WeScience account for some 14,000 utterances—with a fairly dense distribution up to about 50 tokens in length—and approximately 260,000 tokens. For convenience, the corpus is consecutively distributed across 16 files ('segments'), of which the last three are at present reserved as held-out test data. Since September 2009, a Redwoods treebank of the first 13 segments has been available, comprising a total of 11,500 utterances (where well below one percent of utterance boundaries required manual revision). Of these, 88% could be analyzed by the grammar, but in a little more than nine percent of all cases the annotator rejected all available analyses. With some 9,100 remaining gold-standard analyses, current treebank coverage in WeScience is just below 80%.

4. Scaling Up: The Complete Wikipedia

Article Extraction Wikipedia regularly releases complete database dumps, and we start from the 'release' snapshot dated July 2008 (which also was the starting point for the earlier WeScience work). While the dump file contains over seven million article elements, the majority of these are redirects (alternate names) and non-encyclopedic content (help pages, meta discussions, templates, images, and others).[8] In extracting plain text article sources, we ignore such entries. Furthermore, we skip over articles of less than 2000 characters (typically mere cross-references) and those that are part of the WeScience sub-corpus.

Our resulting WikiWoods collection counts approximately 1.3 million content articles, which need to be preprocessed, segmented, and parsed. We have completed the first two of these steps (using a high-performance compute cluster), which resulted in about 55 million utterances—each of about 16.3 tokens average length—and a file set of approximately 2.3 gigabytes, when compressed.

[6] We found the open-source tool *tokenizer* to work best for our purposes; see `http://www.cis.uni-muenchen.de/~wastl/misc/`.

[7] See `http://www.delph-in.net/wescience/` for additional information and download instructions.

[8] These are articles whose names start in *Category:*, *Help:*, *Image:*, *MediaWiki:*, *Portal:*, *Template:*, or *Wikipedia:*.

$$\begin{array}{l}
\langle h_1, \\
\left|\begin{array}{l}
h_3{:}\_\mathrm{the\_q}(x_5, h_6, h_4), h_7{:}\_\mathrm{song\_n\_of}(x_5\{\mathrm{PERS}\ \mathit{3}, \mathrm{NUM}\ \mathit{sg}\}, \_), \\
h_9{:}\_\mathrm{later\_a\_1}(\_, e_2), h_9{:}\_\mathrm{cover\_v\_1}(e_2\{\mathrm{SF}\ \mathit{prop}, \mathrm{TENSE}\ \mathit{past}, \mathrm{MOOD}\ \mathit{ind}\}, x_{11}, x_5), \\
h_{16}{:}\mathrm{compound\_name}(\_, x_{11}, x_{17}), \\
h_{19}{:}\mathrm{proper\_q}(x_{17}, h_{20}, h_{21}), h_{22}{:}\mathrm{named}(x_{17}\{\mathrm{PERS}\ \mathit{3}, \mathrm{NUM}\ \mathit{sg}\}, \mathit{Harry}), \\
h_{13}{:}\mathrm{proper\_q}(x_{11}, h_{14}, h_{15}), h_{16}{:}\mathrm{named}(x_{11}\{\mathrm{PERS}\ \mathit{3}, \mathrm{NUM}\ \mathit{sg}\}, \mathit{Nilsson})
\end{array}\right| \\
\{ h_{20} =_q h_{22}, h_{14} =_q h_{16}, h_6 =_q h_7 \} \rangle
\end{array}$$

Figure 2: Semantic representation (compare to Figure 1). The details of underspecification are not important here, but note that the arguments of the passive are adequately recovered.

Deep Parsing: Setup As sketched above, we produce the syntactic and semantic annotation for each utterance, usually but not always a sentence, by parsing the utterance using a relatively efficient parser and an HPSG-based grammar, recording the most likely resulting analysis according to a statistical model trained on a manually treebanked subset of the Wikipedia. We use the open-source PET chart parser (Callmeier, 2000), which includes support for a non-trivial set of preprocessing rules for token-based normalization (Adolphs et al., 2008) included as part of the English Resource Grammar. This mechanism accommodates on-the-fly lexical entries for lightweight named entities, as recognized by string-level patterns, including numerals, dates, URLs, measure phrases; furthermore, unknown words are treated in a similar fashion, based on part-of-speech tags.

The ERG provides a manually constructed lexicon of some 35,000 entries, designed to include all closed-class words of the language as well as most verbs of reasonable frequency and most of the syntactically idiosyncratic nouns, adjectives and adverbs of the language. Thus standard POS tags are generally sufficient to construct for unknown words valid on-the-fly lexical entries that do not compromise the linguistic accuracy of the resulting analyses. The preprocessing rules of the grammar add lexical edges to the chart, augmenting the inventory of edges supplied by the existing lexicon for all known words in the utterance to be parsed.

In the main parsing phase, the PET parser applies the roughly 200 phrasal constructions defined in the ERG to the parse chart produced by the preprocessing phase, and pursues an all-paths, bottom-up chart parsing strategy, using the single combinatory operation of unification to construct all and only those phrases which satisfy the constraints of the typed feature structures comprising both rules and lexical entries. The parser produces a packed forest of the analyses licensed by the grammar, and can then unpack these analysis in order of likelihood (Oepen & Carroll, 2000). When preparing the manually annotated WeScience sub-corpus in order to create training data for our parse-ranking model, we recorded all of the candidate analyses (up to a practical limit of 500 per utterance), whereas the parser is only asked to produce the single most likely analysis when we are parsing the full Wikipedia. For the treebanking phase, we employed the [incr tsdb()] platform for grammar competence and performance profiling (Oepen, 2001), which includes a sophisticated graphical tool for the task of disambiguation, enabling annotators to efficiently and consistently identify the intended analysis from among the candidates in the parse forest.

While the ERG is designed to be domain-independent, it is inevitable that every new corpus to which the grammar is applied will present some number of previously unaddressed linguistic phenomena in sufficient frequency to merit some focused elaboration of the grammar. Often such phenomena have already been observed in other treebanked corpora such as the Norwegian tourism-oriented LOGON corpus (Lønning et al., 2004) but had not yet risen to the top of the grammarian's to-do list. The Wikipedia corpus is no exception, and the sentence-by-sentence analysis of the WeScience sub-corpus led to several grammar modifications for improved linguistic coverage, including both pre-processing rules and grammatical constructions.

One example of this Wikipedia-motivated grammar elaboration is a rule to admit noun-modifying phrases that consist of a (sometimes hyphenated) noun followed by either an adjective or a passive verb, as in "*context sensitive* grammars" or "*computer-implemented* algorithms". This construction is both highly productive and compositionally transparent, and occurs 250 times in the 10,000-utterance WeScience treebank, but had previously been patched via manually constructed multi-word lexical entries, an approach that obviously would not scale up to the Wikipedia corpus.

Interfacing Markup and Grammar As pointed out earlier, in some cases markup properties also directly affect linguistic analysis. A prominent such example is the use of italics in a function similar to quotation, viz. drawing a use – mention distinction or demarcating foreign language material (examples (1) and (2) below, respectively). Specifically, observe that (1) would be ungrammatical if read literally, due to the lack of a determiner in the final noun phrase.

(1) For example, in the following example, *one* can stand in for *new car*.

(2) The above example in German would be *Ein Mann beißt den Hund* or *Den Hund beißt ein Mann*.

To enable correct grammatical analysis of such examples, the scope of italics markup needs to be made accessible to the parsing system. To abstract from the concrete syntax of a specific markup language (like Wikipedia, HTML, or LaTeX), we have started to augment the ERG analysis grammar with selected elements of an *abstract* Grammatical Markup Language (dubbed GML). During input preprocessing for the parser, for example, italicized words or phrases are enclosed in 'opening' and 'closing italics' tokens: ¦*i new car i*¦, for part of example (1) above.

Parser-internally, GML tokens like these are treated much like punctuation marks; i.e. in the approach of Adolphs et al. (2008) such tokens are re-combined with adjacent 'regular' tokens (i.e. non-markup and non-punctuation ones) and then syntactically analyzed as pseudo-affixes.

This approach has the benefit of eliminating attachment ambiguities for punctuation and markup tokens (they always attach lowest, i.e. lexically), and furthermore it yields a perfect predictor of standard whitespace conventions around punctuation marks—commas, for example, are pseudo-suffixes; opening parentheses, on the other hand, are pseudo-prefixes. Aligning the treatment of some markup with the existing analysis of punctuation provides a fruitful starting hypothesis for our WeScience experiments. In the case of italicized phrases, the grammar is thus enabled to apply its existing apparatus for 'recognizing' quoted expressions (as an uninterpreted, strictly left-branching binary tree). Once a complete, properly bracketed phrase is recognized, unary rules map the corresponding constituent into a suitable syntactic category, typically a phrase with syntactic properties closely resembling a proper name.

Preliminary Quality Evaluation As a first test of the infrastructure and methodology for parsing and recording results on the full corpus, we selected a random subset of articles containing a little more than half a million utterances for study. Using a statistical model for parse ranking trained on the WeScience Treebank, we applied the ERG to this subset of the corpus, recording only the one most probable parse (if any) for each utterance. At just below 85%, grammatical coverage on this (more diverse) data is comparable to, though slightly lower than, the earlier WeScience experiment.

In order to obtain a rough measure of the effectiveness of statistical parse disambiguation, we randomly selected 1,000 parsed sentences from this subset corpus, and carried out a manual evaluation (using the principal ERG developer as an expert, if potentially mildly biased judge) of the quality of the top-ranked analyses assigned by the grammar. Excluding a handful of items which suffered from incorrect sentence segmentation or typographical errors, each parse was judged to be of one of three qualities: *correct*, *nearly correct*, or *incorrect*. For a parse to be judged correct, every aspect of the analysis had to be fully adequate, including both syntax and semantics. Those items judged as nearly correct contained one or at most two minor errors which did not materially affect the overall meaning of the utterance; the errors were typically misbracketing within a complex nominal compound, misattachment of a modifying prepositional phrase, or an infelicitous coordination bracketing. If an analysis contained more than two such minor errors, or a more serious error resulting in substantial damage to the meaning of the utterance, the parse was judged to be incorrect. Clearly, longer items typically present more opportunities for error, so it is to be expected that item length will correlate with parse quality using this coarse-grained method of evaluation. The results of this initial evaluation on 1000 items are summarized in Table 1.

Table 1: Overview of manual parse quality evaluation for 1,000-utterance sample.

Item Length	**Incorrect Parse**	**Nearly Correct**	**Correct Parse**	**Total Items**
1 – 4	3	10	250	265
5 – 14	44	49	237	333
15 – 24	50	71	123	248
$\geq$ 25	50	51	47	154
Totals	**147**	**181**	**657**	**1000**

Our preliminary manual evaluation suggests that the quality of the analyses assigned fully automatically is quite good, with more than 83% of the analyses judged as correct or nearly correct, and more than two-thirds judged as fully correct. The percentage of incorrectly analyzed items rises as sentence length increases, as expected, but even for sentences of 5 to 24 tokens in length, 82.6% of these received nearly or fully correct analyses. There is still, of course, substantial room for improvement in statistical disambiguation here, for example re-training the parse selection model on a larger hand-constructed Redwoods treebank, or trying to self-train on much larger amounts of data. Error analysis also is needed on the 15% of items which received no parse at all, to guide improvements both in our preprocessing techniques and in the grammar, in order to increase the observed coverage as we move to parsing the full corpus. Based on trial runs of about ten percent of the total text, we estimate the cost of parsing the full WikiWoods corpus at about 100,000 cpu hours. The project has access to a 6000-core HPC installation at the University of Oslo, and we estimate that parsing can be completed in about ten days.

5. Discussion—Outlook

We prepare the WikiWoods collection in the hope that the depth of available information and sheer scale of the resource will make it attractive for NLP tasks such as lexical semantic acquisition or ontology learning. Due to the parsing-centric nature of our approach, the initial WikiWoods release in May 2010 will contain about 15% gaps in treebank coverage, still resulting in some 47 million annotated utterances, which ideally will be validated through community adaptation. In subsequent work, we plan to adapt the robust parsing approach of Zhang & Kordoni (2008) to mitigate this coverage shortfall. Specifically, we expect to combine their method of robustly producing so-called 'pseudo-derivations' (that are not strictly speaking consistent with the full constraints of the grammar) with a technique for robust semantic composition, so as to also be able to obtain an MRS meaning representation, even where there is no full HPSG feature structure.

Acknowledgements

This report on work in progress owes a lot to prior investigation by Woodley Packard, who started parsing Wikipedia using the ERG as early as 2003. We are furthermore indebted to Peter Adolphs, Francis Bond, Yusuke Miyao, Jan Tore Lønning, Erik Velldal, and Yi Zhang for their encouragement and productive comments. This work is in part funded by the University of Oslo, through its research partnership with the Center for the Study of Language and Information at Stanford University. Experimentation and engineering on the scale of Wikipedia is made possible though access to the TITAN high-performance computing facilities at the University of Oslo, and we are grateful to the Scientific Computation staff at UiO, as well as to the Norwegian Metacenter for Computational Science.

References

Adolphs, P., Oepen, S., Callmeier, U., Crysmann, B., Flickinger, D., & Kiefer, B. (2008). Some fine points of hybrid natural language parsing. In *Proceedings of the 6th International Conference on Language Resources and Evaluation.* Marrakech, Morocco.

Callmeier, U. (2000). PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering, 6 (1) (Special Issue on Efficient Processing with HPSG),* 99 – 108.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering.* Madrid, Spain.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction. *Journal of Research on Language and Computation, 3*(4), 281 – 332.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering, 6 (1),* 15 – 28.

Lønning, J. T., Oepen, S., Beermann, D., Hellan, L., Carroll, J., Dyvik, H., Flickinger, D., Johannessen, J. B., Meurer, P., Nordgård, T., Rosén, V., & Velldal, E. (2004). LOGON. A Norwegian MT effort. In *Proceedings of the Workshop in Recent Advances in Scandinavian Machine Translation.* Uppsala, Sweden.

Oepen, S. (2001). [incr tsdb()] — *Competence and performance laboratory. User manual* (Technical Report). Saarbrücken, Germany: Computational Linguistics, Saarland University.

Oepen, S., & Carroll, J. (2000). Ambiguity packing in constraint-based parsing. Practical results. In *Proceedings of the 1st Conference of the North American Chapter of the ACL* (pp. 162 – 169). Seattle, WA.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation, 2*(4), 575 – 596.

Ytrestøl, G., Flickinger, D., & Oepen, S. (2009). Extracting and Annotating Wikipedia Sub-Domains. Towards a New eScience Community Resource. In *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories.* Groningen, The Netherlands.

Zhang, Y., & Kordoni, V. (2008). Robust parsing with a large HPSG grammar. In *Proceedings of the 6th International Conference on Language Resources and Evaluation.* Marrakech, Morocco.