The WeSearch Corpus, Treebank, and Treecache

A Comprehensive Sample of User-Generated Content

Jonathon Read♣, Dan Flickinger♡, Rebecca Dridan♣, Stephan Oepen♣, and Lilja Øvrelid♣

♣ University of Oslo, Department of Informatics

♡ Stanford University, Center for the Study of Language and Information

{jread|rdridan|oe|liljao}@ifi.uio.no, danf@stanford.edu

Abstract

We present the WeSearch Data Collection (WDC)—a freely redistributable, partly annotated, comprehensive sample of User-Generated Content. The WDC contains data extracted from a range of genres of varying formality (user forums, product review sites, blogs and Wikipedia) and covers two different domains (NLP and Linux). In this article, we describe the data selection and extraction process, with a focus on the extraction of linguistic content from different sources. We present the format of syntacto-semantic annotations found in this resource and present initial parsing results for these data, as well as some reflections following a first round of treebanking.

Keywords: User-Generated Content, Open-Source Corpus, Manually Validated Treebank, Automatically Created Treecache

1 Background—Motivation

An ever increasing proportion of the Internet is comprised of so-called User-Generated Content (UGC). Applications that seek to 'tap into the wisdom of the masses' demand various levels of natural language processing (NLP) of these kinds of text. For statistical parsing, for example, Foster et al. (2011) observe that common off-the-shelf parsers—trained on the venerable Wall Street Journal data—perform between ten and twenty F_1 points worse when applied to social media data. To enable more R&D into the linguistic properties of common types of UGC as well as into the technological challenges it presents for NLP, we are making available a freely redistributable, partly annotated, comprehensive sample of UGC—the WeSearch Data Collection (WDC).

The term 'domain adaptation' has at times been used to characterise the problem of tuning NLP tools for specific types of input (Plank, 2011). In our view, however, it is desirable to reserve the term 'domain' for *content* properties of natural language samples (i.e. the subject matter), and complement it with the notion of 'genre' to characterise *formal* properties of language data (i.e. the text type). On this view, *parser adaptation* would typically comprise both domain and genre adaptation (and possibly others). Thus, in the WDC resource discussed below, we carefully try to tease the two dimensions of variation apart—seeking to enable systematic experimentation along either of the two dimensions, or both.

In this work, we develop a large, carefully-curated sample of UGC comprised of three components: the WDC Corpus, Treebank, and Treecache. Here, the corpus comprises the unannotated, but utterance-segmented text (at variable levels of 'purification'); the treebank provides fine-grained, gold-standard syntactic and semantic analyses; and the treecache is built from automatically constructed (i.e. not manually validated) syntacto-semantic annotations in the same format.[1]

The article is structured as follows: Section 2 describes the selection of sources for the data collection and Section 3 goes on to detail the harvesting and extraction of content from these data sources. In Section 4 we describe the organisation of the corpus into three versions with different format, as well as organisation with respect to genre/domain and standardised train-test splits. Moving on to annotation, Section 5 presents the annotation format for the data collection, while Section 6 describes initial parse results for the full data collection and Section 7 provides some reflections regarding quality and domain-and genre specific properties of the data following an initial round of treebanking. Finally, Section 8 details next steps in terms of corpus refinement and ultimate release of the resource.

2 Data Selection

When selecting data for our corpus, we are firstly interested in a variety of registers of user-generated content (i.e. genres, in our sense) that represent a range of linguistic formality. To date, we therefore obtained text from user forums, product review sites, blogs, and Wikipedia. Albeit 'user-generated' only in a stretch, future versions of the corpus will also include open-access research literature. Secondly we acquired text from sources that discuss either the Linux operating system or natural language processing. The choice of these domains is motivated by our assumption that the users of the corpus will be more familiar with the language used in connection with these topics than (for example) that used in the biomedical domain.

Table 1 lists the complete set of data sources for the first public release of the WDC.[2] The selection reflects linguistic variation (ranging from the formal, edited language of Wikipedia and blogs, to the more dynamic and informal

[1] In recent literature, the term *treebank* is at times used to refer to automatically parsed corpora. To maintain a clear distinction between validated, gold-standard vs. non-validated annotations, we coin the parallel term *treecache* to refer to automatically created collections of valuable, if not fully gold-standard trees. Note that this notion is related to what Riezler et al. (2000), in the context of Lexical Functional Grammar, dub a *parsebank*—though not fully identical to the interpretation of Rosén et al. (2009) of that term (also within the LFG framework).

[2] See www.delph-in.net/wesearch for technical details and download instructions.

Domain	Genre	Source(s)	Format
NLP	Wikipedia	WeScience (Ytrestøl et al., 2009)	Wikitext

NLP	Blogs	`blog.cyberling.org`
		`gameswithwords.fieldofscience.com`
		`lingpipe-blog.com`
		`nlpers.blogspot.com`
		`thelousylinguist.blogspot.com`
			HTML

Linux	Wikipedia	`www.wikipedia.org`	Wikitext

Linux	Blogs	`embraceubuntu.com`
		`www.linuxscrew.com`
		`www.markshuttleworth.com`
		`www.ubuntugeek.com`
		`ubuntu.philipcasey.com`
		`www.ubuntu-unleashed.com`
			HTML

Linux	Software reviews	`www.softpedia.com/reviews/linux/`	HTML
Linux	User forums	The *Unix & Linux* subset of the April 2011 Stack Exchange Creative Commons Dump.	HTML

Table 1: Sources of user-generated content in the WDC 1.0.

language of product review sites and user forums) and establishes a broad domain of information technology—in order to extend and complement prior related initiatives (Bird et al., 2008; Baldwin et al., 2010; Flickinger et al., 2010; Foster et al., 2011; Schäfer et al., 2011). To make redistribution of the final corpus as straightforward as possible we attempted to select sources that were published under an open licence. Where this was not possible we contacted the authors for redistribution permission. Thus, the WDC is a completely *open-source* resource.

3 Harvesting and Extraction

Sampling user-generated content from potentially noisy on-line sources necessitates the interpretation of various markup formats, as well as the determination of which components of a Web page, say, actually correspond to *relevant* and *linguistic* content (as opposed to, for example, navigational elements, meta-information, rigidly structured, or non-linguistic data). The paragraphs below describe this process for each data format.

The blogs and reviews are written in HTML, with document structure and formatting conventions varying for each data source. Despite this, however, it is possible to extract content in a fairly generic manner, given the following information for each source:

1. A regular expression that extracts the title of a web page. We did not extract titles from the HTML body of the page as their representation (and even presence) was not consistent across sources. Instead we use the title from the HTML header, after first stripping automatically-generated text such as the blog title.

2. A regular expression that identifies the *start* tag of the post body. It is not feasible to define a pattern that extracts the entire body due to the possibility of nesting elements. Instead the start of the post is identified, then the subsequent text is searched until the closing tag is found.

3. A function that removes superfluous, automatically-generated text from the post body. In most cases this was not necessary, but some blogs include automatically-generated text, such as author identification messages, links to related posts, or links to share posts on social networks. We remove such content as it is not representative of the linguistic content.

Obtaining text from posts in the Stack Exchange forums is much more straightforward, as the HTML is effectively sanitised by the data dump, being stored as entries in a database. Thus, content is easily extracted from the appropriate columns.

Text from the blogs, reviews and forums was segmented using the sentence splitter from the Stanford CoreNLP Tools.[3] These tools record token positions using character offsets, enabling us to track sentence positions, and thus record pointers from corpus items to positions in the source HTML documents. This means that (a) there is full accountability in terms of extractions from and modifications to the raw source files and (b) a parser's output may be used to annotate the source files—as done, for example, by the ACL Anthology Searchbench (Schäfer et al., 2011).

To complete the preprocessing a number of steps were performed in parallel. These include: removal of superfluous whitespace; removal of tables and comments; replacing `img` and `code` elements with placeholders; and ensuring segmentation following sentence-breaking tags.[4] We retain a small subset of elements[5] which we believe to be informative for subsequent processing (see Section 4). Such elements are preserved, but their attributes are removed (e.g. the `href` attributes of `a` elements). Better elimination of some non-linguistic content (e.g. displayed code snippets or formulae that do *not* form a constituent in a sentence) re-

[3] Available at `www-nlp.stanford.edu/software/corenlp.shtml`.

[4] Specifically, instances of `br`, `div`, `li`, `p` or `pre`.

[5] Specifically, instances of `a`, `b`, `em`, `h1`, `h2`, `h3`, `kbd`, `li`, `s`, `small`, `strong`, `sub` or `sup`.

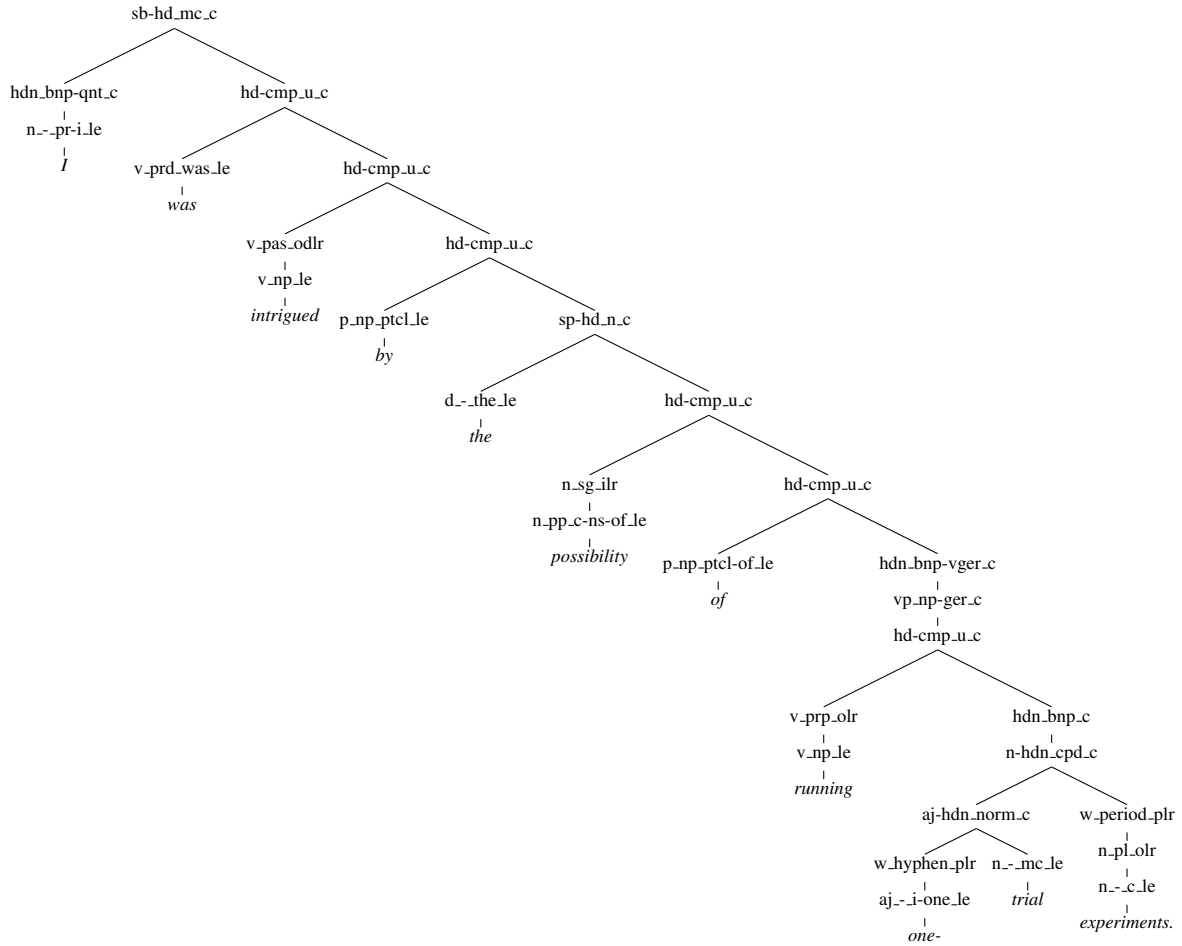

Figure 1: Syntactic representation for *I was intrigued by the possibility of running one-trial experiments.* The HPSG derivation is labelled with identifiers of grammatical constructions and types of lexical entries.

mains an open challenge, to achieve a good balance of text coherence and 'purity'.

For the Wikipedia sections, a different preprocessing process was used. To extract a coherent, domain-specific part of Wikipedia we followed the established methodology and toolchain of Ytrestøl et al. (2009). For the Linux-related WDC sub-corpus, sub-domains are approximated by creating a seed set from relevant items in the Wikipedia category system. We then performed a connectivity analysis of articles linked to by the seed set, and discard those that are infrequently referenced. Once linguistic content had been extracted, we segmented the text into 'sentence' (or other top-level utterance type). Again following Ytrestøl et al. (2009), for this first public release of the WDC we combined the open-source tokeniser package[6] with a handful of heuristic segmentation (or in a few cases, anti-segmentation) rules that take advantage of surface layout properties.

4 Corpus Organisation

In extracting relevant content, some aspects of layout can actually affect linguistic analysis (and thus it can be desirable to keep some markup in the corpus): For example, knowing that a segment of text comes from a heading or an itemised list might activate special syntactic analyses, emphasised text may indicate foreign or otherwise, and forced geometric layout can interact with sentence segmentation. Thus, we provide the WDC data in three distinct formats, viz. as (*L0*) the raw source files; (*L1*) the linguistic content only, but with most markup preserved in the original syntax (HTML or Wikitext); and (*L2*) the same textual content, but with markup elements considered potentially relevant to linguistic analysis normalised to a format-independent scheme.

The L0 version is organised as subsets of domain and genre collections. Each subset contains a directory for each source; files within the directory correspond to the path on the source site.

The L1 and L2 organisation is also by domain and genre, but differs in that texts are collected into sections of approximately 1,000 sentences. Each section contains articles from a single source web site, and articles are not split across sections. For each domain and genre collection (except NLP Wiki[7]) we reserve sections as follows: held-out

[6] See `www.cis.uni-muenchen.de/~wastl/misc/` for background.

[7] A different numbering convention is used for the NLP Wiki collection because it is derived directly from previous work (Ytrestøl et al., 2009). In this collection: 01–12 is for training, 13 is for testing and 14–16 is held-out.

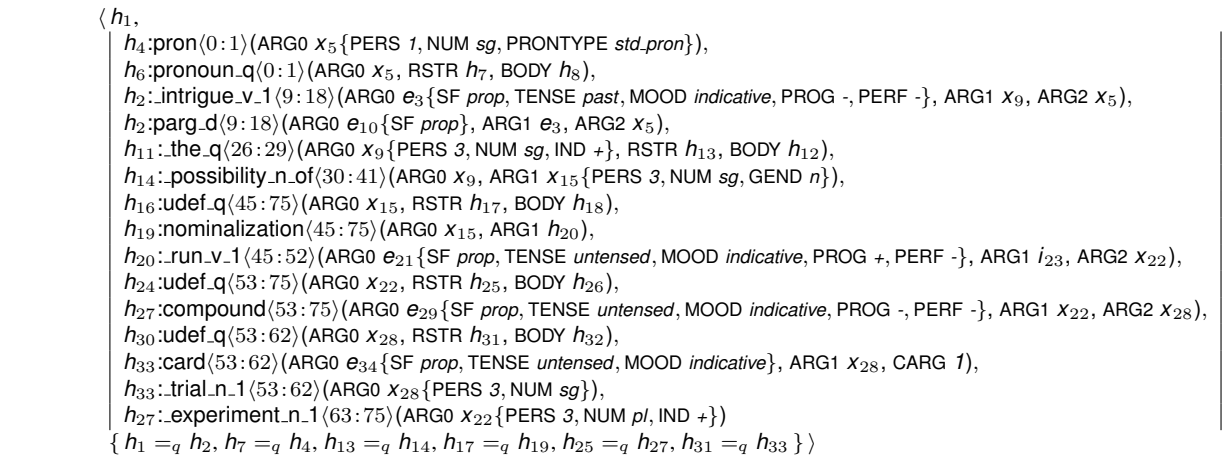

Figure 2: Semantic representation of our running example (compare to Figure 1). The details of underspecification are not important here, but note that the arguments of the passive are adequately recovered.

data (00, 01); test data (02, 03); and training data (04 and upwards). Test data is randomly drawn from the collection, except when (as for Linux and NLP blogs) there are multiple sources in a collection, in which case Section 02 is instead drawn from a single source which is not represented in the training sections. The rationale for these single source test sets is based on one potential use case for this data collection, which is as a resource for exploring domain and genre effects. These single source test sets could enable researchers to test whether a model could be learnt for the 'blog' genre that would generalise to truly unseen text. The mixed source test sets are based on the conventional idea that test and training data are drawn from the same distribution.

Articles in L1 and L2 are organised chronologically where this makes sense (i.e. for blogs, reviews, and forums), and identified using an eight-digit code. This identifier is accompanied by a pointer to the L0 source file. Each item is accompanied by a character offset that points to its location in the source file, and a list of character offsets that represent deletions made in the cleaning process described in Section 3. Wikipedia, as an encyclopedia, doesn't have a chronological order to its articles, and so utterance identifiers were just assigned sequentially.

5 Format of Annotations

The type of syntacto-semantic annotations available both in the WDC Treebank and Treecache follows the best practices of the WeScience Treebank and WikiWoods Treecache (Ytrestøl et al., 2009; Flickinger et al., 2010) and is exemplified by Figures 1 and 2. The annotation is obtained from a broad-coverage parsing system couched in the HPSG framework—the LinGO English Resource Grammar (ERG; Flickinger, 2002). Internally, each full HPSG analysis is characterised by the derivation tree (in Figure 1), labelled 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, subcategorisation information, other grammaticalised 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 among the nodes of Figure 1.

A more conventional representation of syntactic information is available in the form of constituent trees labelled with 'classic' category symbols (not shown here), 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 customisation 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 labelling process, i.e. obtain a custom set of constituent trees from the original derivations.

In terms of semantic annotation available in the WDC, 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-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* relation provides the bracketing of the complex noun phrase; however, syntax does not necessarily determine the exact functional structure of complex nominals, hence the underspecified relation in this case. Finally, observe how at the level of semantics the role assignments

Domain	Genre	Section(s)	Items	Length	Types	Coverage	Resource Exhaustion
Linux	Blog	Held out (00, 01)	1966	14.4	5898		
		Test (single source) (02)	1577	13.5	3976	85.8%	4.1%
		Test (mixed source) (03)	1074	14.2	3760	83.4%	4.7%
		Train (04–65)	57599	12.9	47443	82.2%	4.1%
NLP	Blog	Held out (00, 01)	1969	17.8	6763		
		Test (single source) (02)	659	20.0	3032	81.3%	10.9%
		Test (mixed source) (03)	994	18.0	4232	83.1%	7.2%
		Train (04–42)	36104	17.1	36778	83.4%	5.8%
Linux	Wiki	Held out (00, 01)	1583	19.2	4842		
		Test (02, 03)	1894	19.2	6342	86.2%	9.9%
		Train (04–45)	37263	18.5	50228	85.4%	9.6%
NLP	Wiki	Held out (14–16)	2412	20.0	9426		
		Test (13)	1001	17.6	4570	86.9%	8.2%
		Train (01–12)	10557	18.1	10173	87.0%	7.4%
Linux	Forum	Held out (00, 01)	2051	14.7	4771		
		Test (02, 03)	2007	14.6	4659	78.9%	3.0%
		Train (04–65)	53160	14.5	31816	79.7%	2.9%
Linux	Review	Held out (00, 01)	1994	21.1	5028		
		Test (02, 03)	2010	19.8	4955	80.6%	6.6%
		Train (04–13)	9667	19.1	10976	81.4%	5.6%

Table 2: Various corpus statistics, in terms of counts, average length, vocabulary size, and initial parsability.

are normalised: 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 analysed by the grammar, e.g. the dative shift in *Kim gave Sandy a book.* vs. *Kim gave a book to Sandy.*

6 Initial Parsing Results

In order to get some idea of the parsability of the data, we parsed all test and training sections of the data in the L1 version of the data collection. We used the PET parser (Callmeier, 2000) and the 1111 release of the ERG, with additional HTML markup handling rules added to the preprocessing stage. No other adaptation was made to the standard parser for this run. Statistics for each section are shown in Table 2, including the number and average length of the items, the number of unique, non-punctuation tokens and what percentage of the items the parser was able to analyse.

We see that the highest parse coverage is over the Wikipedia text, which follows both from the more formal register and possibly because the grammar has been previously adapted to the NLP Wikipedia text. Indeed, the majority of parse failures in this genre are caused by resource exhaustion, due to setting a time and memory limit in the parser. Parse coverage numbers for the blog and review genres are over 80 %, with higher coverage over Linux blogs mostly related to the shorter average utterance length, which results in less time outs. The Linux Forum sections stand out here as the most difficult text to parse. While the drop in coverage is not so large, we would expect higher coverage over such short utterances, since (as can be seen), resource limitation is less of a problem. An initial examination of the unparsed utterances showed that some of the issues were caused by ungrammatical utterances, or sentence fragments possibly caused by imperfect segmentation. Other problems however were domain-specific, including example commands within a sentence, or domain-specific senses of common words such as `make`, `screen` and `bash`.

Differences in domain are most often explained as lexical differences (Rimell & Clark, 2008; Hara et al., 2007). To get some idea of the lexical properties of the corpus section, we used the methods of Rayson & Garside (2000) to find those lexical items most distinctive of the various sections. While many such terms are predictable (*ubuntu*, *install*, *sudo* for Linux versus *language*, *words*, *model* for NLP), there are some unexpected trends. One example is the strong preference for first person pronouns (*I*, *we*) in the NLP blogs contrasting with a higher than expected proportion of *you* across the Linux blogs. Opportunities for further exploration of such effects, as well as non-lexical differences, are some of the benefits we anticipate from this corpus.

7 Initial Treebanking Reflections

To get a first impression of the quality of the WDC text and domain- and genre-specific properties, we manually inspected and corrected the segmentation produced for one section of the corpus, and then performed a first round of grammar-based treebanking. The automatic utterance segmentation of section WNB03 (the mixed-source test data in the NLP blogs) produced 994 items, but manual line-by-line correction resulted in 1078 items, an 8 % undercount for this one section. Many of the overlooked sentence boundaries were masked by HTML markup, or by apparently unexpected punctuation clusters, question marks, and unconventional spacing. Less frequently, spurious segmen-

tation breaks were introduced, following punctuation marks mistaken for clause boundary marks such as the exclamation mark that is part of the name *Yahoo!* or the final period in one common spelling of *Ph.D.* in mid-sentence. We expect to investigate improved sentence boundary detection, to be attentive to the more frequent of these sentence-boundary conditions that we find in the UGC data; but we will also manually correct the segmentation for the portion of the corpus to which we add gold-standard linguistic annotation.

Our method for assigning syntactic and semantic annotations is the same one presented in Ytrestøl et al. (2009), where we parse each sentence using the ERG (see Section 5 above), record the most likely analyses (up to a maximum of 500), and then manually select the correct analysis from this parse forest, recording both the derivation tree and its associated semantic representation. Since we expect user-generated language data to present linguistic and orthographic variation different from the largely edited texts previously considered in ERG development, the early rounds of annotation will also help us to identify opportunities for improvements in the grammar, leading to more than one cycle of annotation for at least these early sections.

For an initial profile of the behaviour of the grammar on this kind of data, we manually annotated the section of the corpus for which we had already hand-corrected the sentence segmentation, and made several observations. Some of the necessary additions to the grammar are unsurprising for this genre of text, such as missing lexical entries for emoticons (such as '*:)*' or '*:-)*' or '*:P*'), exclamations ('*D'oh!*' or '*ah-ha*'), and abbreviations (e.g. '*btw*', '*omg*', or '*imho*'). Similarly genre-specific informal expressions such as *the likes of [...]* and *crammed in some [...]* also appeared in the course of this annotation exercise, as did some domain-specific entries such as the Linux-forum verb *gc-ed* ('garbage-collected'). Somewhat to our surprise, we did not find very many grammatically significant typographical errors in this formally unedited text: only in 27 of the 1078 items did we find grammatically relevant authored errors, including lexical substitutions (e.g. *is* for *in*), omissions (missing articles), insertions (e.g. *could could*), or scrambled word order (e.g. *defined by bag the*).

After making the obvious improvements to the ERG lexicon and then running through the annotation cycle (parse – update – treebank – tune) a few times, we ended up with just over 80 % of the 1051 well-formed items in this 1078-item section of the corpus receiving a manually confirmed correct analysis. We have not yet put any effort into more substantial improvements to the grammar that may be motivated by syntactic phenomena of unusual frequency in this corpus (such as the creative and enthusiastic use of parentheticals), but we expect further improvements in annotated coverage to result from working with this corpus.

8 Discussion—Outlook

The next steps in developing the WeSearch Data Collection (WDC) include a round of parser adaptation, including tuning of the statistical parse selection model and potentially some grammar modification to address the special features of the different genre. As part of that tuning, we will need to address genre- and domain-specific challenges that came up in the exploratory processing—often relating to the interface of markup interpretation and linguistic analysis. One issue in particular is how technical material such as source code, formulae and configuration file snippets should be (a) detected, and (b) handled. Manual verification of segmentation and analysis over subsets of the data is another stage which we will perform before the first public release of the corpus, planned for May 2012.

Finally, we release the WDC in the hope that it may contribute to a growing pool of community resources for further research into the information available in user-generated content. We hope that such a curated corpus, separated by domain and genre, and with deep syntacto-semantic analyses will allow research beyond that which currently available resources permit.

Acknowledgements

We are grateful to our colleagues at the Oslo Language Technology Group for fruitful discussions and suggestions, as well as to three anonymous reviewers for insightful comments. This work is in part funded by the Norwegian Research Council through its VerdIKT programme. Large-scale experimentation and engineering is made possible though access to the TITAN high-performance computing facilities at the University of Oslo, and we are grateful to the Scientific Computating staff at UiO, as well as to the Norwegian Metacenter for Computational Science and the Norwegian tax payer.

References

Baldwin, T., Martinez, D., Penman, R., Kim, S. N., Lui, M., Wang, L., & MacKinlay, A. (2010). Intelligent Linux information access by data mining: the ILIAD project. In *Proceedings of the NAACL 2010 workshop on computational linguistics in a world of social media: #SocialMedia* (pp. 15–16). Los Angeles, USA.

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., & Tan, Y. F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the 6th International Conference on Language Resources and Evaluation* (pp. 1755–1759). Marrakech, Morocco.

Callmeier, U. (2000). PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, *6*(1), 99 – 107.

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. (2002). On building a more efficient grammar by exploiting types. In S. Oepen, D. Flickinger, J. Tsujii, & H. Uszkoreit (Eds.), *Collaborative language engineering* (pp. 1–17). Stanford: CSLI Publications.

Flickinger, D., Oepen, S., & Ytrestøl, G. (2010). Wikiwoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation* (pp. 1665–1671). Malta.

Foster, J., Cetinoglu, O., Wagner, J., Roux, J. L., Nivre, J., Hogan, D., & Genabith, J. van. (2011). From news to comment. Resources and benchmarks for parsing the language of Web 2.0. In *Proceedings of the 5th International Joint Conference on Natural Language Processing.* Chiang Mai, Thailand.

Foster, J., Cetinoğlu Özlem, Wagner, J., & Genabith, J. van. (2011). Comparing the use of edited and unedited text in parser self-training. In *Proceedings of the 12th International Conference on Parsing Technology (IWPT 2011)* (pp. 215–219). Dublin, Ireland.

Hara, T., Miyao, Y., & Tsujii, J. (2007). Evaluating impact of re-training a lexical disambiguation model on domain adaptation of an HPSG parser. In *Proceedings of the 10th International Conference on Parsing Technology (IWPT 2007)* (pp. 11–22). Prague, Czech Republic.

Plank, B. (2011). *Domain adaptation for parsing.* Ph.d. thesis, University of Groningen.

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. In *The workshop on comparing corpora* (pp. 1–6). Hong Kong, China: Association for Computational Linguistics.

Riezler, S., Kuhn, J., Prescher, D., & Johnson, M. (2000). Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics* (pp. 480 – 487). Hong Kong, Hong Kong.

Rimell, L., & Clark, S. (2008). Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)* (pp. 475–484). Honolulu, USA.

Rosén, V., Meurer, P., & Smedt, K. D. (2009). LFGParsebanker. A toolkit for building and searching a treebank as a parsed corpus. In *Proceedings of the seventh international workshop on treebanks and linguistic theories* (pp. 127 – 133). Utrecht, The Netherlands.

Schäfer, U., Kiefer, B., Spurk, C., Steffen, J., & Wang, R. (2011). The ACL Anthology Searchbench. In *Proceedings of the ACL-HLT 2011 system demonstrations* (pp. 7–13). Portland, Oregon, USA.

Ytrestøl, G., Oepen, S., & Flickinger, D. (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.