What I've learned about annotating informal text (and why you shouldn't take my word for it)

Nathan Schneider
School of Informatics
University of Edinburgh
Edinburgh, Scotland, UK
nschneid@inf.ed.ac.uk

1 Introduction

In conjunction with this year's LAW theme, "Syntactic Annotation of Non-canonical Language" (NCL), I have been asked to weigh in on several important questions faced by anyone wishing to create annotated resources of NCLs.

My experience with syntactic annotation of non-canonical language falls under an effort undertaken at Carnegie Mellon University with the aim of building an NLP pipeline for syntactic analysis of Twitter text. We designed a linguistically-grounded annotation scheme, applied it to tweets, and then trained statistical analyzers—first for part-of-speech (POS) tags (Gimpel et al., 2011; Owoputi et al., 2012), then for parses (Schneider et al., 2013; Kong et al., 2014). I will review some of the salient points from this work in addressing the broader questions about annotation methodology.

2 Annotation Scheming

Many annotation schemes have been designed for "canonical" forms of language, such as text in a standard dialect formally edited to meet certain style conventions. In order to annotate non-canonical forms of language, one must determine whether existing schemes should be (a) applied as is, (b) adapted, or (c) avoided in favor of a new scheme. Designing a new annotation scheme is not to be undertaken lightly; on the other hand, if an existing scheme really does not fit the resources, then applying it will likely be a waste of time—because the distinctions it makes are not useful, or because the cost of obtaining the desired number of annotations at the desired level of quality will be too high.

A formula for computing the tradeoffs involved in selecting an annotation scheme would have to involve several variables:

- upfront cost (money, time)—e.g., in writing documentation, building the annotation platform, training annotators
- unit cost (money, time)

interact with

- quality/reliability—will depend on annotator expertise and training, and thoroughness of quality control procedures
- volume
- richness/informativeness—i.e., how many distinctions does the scheme make?
- usefulness/applicability—i.e., how valuable are the annotations for some purpose?

It is clear that higher volume, reliability, and richness will tend to incur higher costs. Usefulness for some downstream application may or may not be clear and measurable during annotation,[1] though frameworks like active learning (Settles, 2012) do take it into explicit consideration to make the annotation process more cost-effective.

We come, then, to the main question: **When is it worth designing a new annotation scheme?** My answer is, *When annotating with an existing scheme would be more painful (costly) than starting afresh.* The second question, **What level of granularity?**, is similarly answered by weighing these tradeoffs: too coarse, and the annotations will not be very informative or useful; too fine, and training annotators will be costly, the annotation will be slow, annotator

[1] And, if a scheme is intended to be general-purpose, usefulness would have to be measured on a battery of tasks to be meaningful.

Proceedings of LAW IX - The 9th Linguistic Annotation Workshop, pages 152–157,
Denver, Colorado, June 5, 2015.

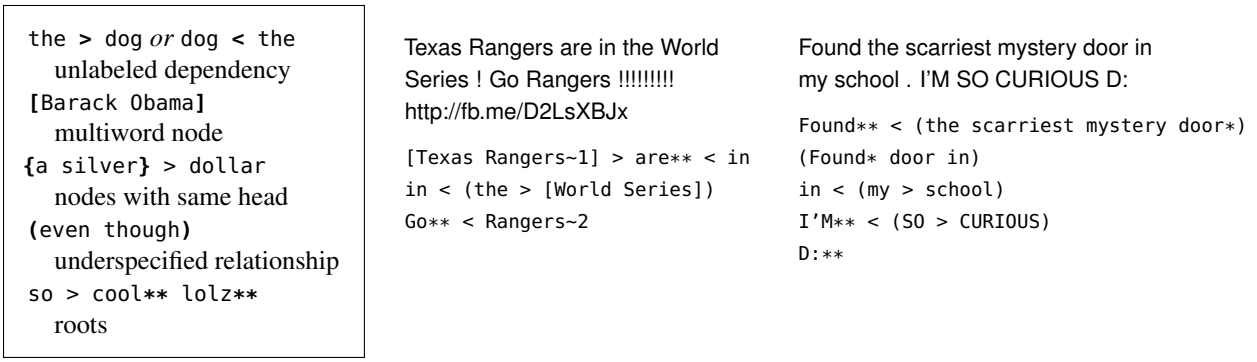

Figure 1: FUDG GFL notation summary and two annotated Twitter examples.

reliability will be low, and some categories may be highly sparse. Estimating these tradeoffs in a particular setting is a qualitative judgment call, so in lieu of a more concrete general principle, I will share some illustrative examples from my own experience.

Twitter POS. Gimpel et al. (2011) introduced (and Owoputi et al., 2012 documented in greater detail) a coarse-grained POS tagset for English tweets. Given that the eventual goal was to build a syntactic parser, we considered extending the Penn Treebank (Marcus et al., 1993) tagset with a few additional tags for social media phenomena (such as emoticons and hashtags). However, we also wanted a "lightweight" tagset to facilitate rapid annotation, and did not feel that the fine-grained inflectional distinctions made in the PTB tags—VB, VBP, VBZ, VBG, VBD, and VBN indicating different forms of verbs, for instance—were an ideal use of annotators' time.

We ultimately decided to craft a tagset coarser grained than the 45 PTB categories, and similar to Petrov et al.'s (2011) "universal" set of 12 categories,[2] but with additional categories suited to tweets: ! (interjection), E (emoticon), U (URL), # (extrasyntactic hashtag), @ (at-mention), and ~ (online discourse marker). Finally, we felt that it would be difficult to force a tokenization of nonstandard words like ima ("I'm going to"), so we opted for a minimal tokenization and added 5 complex tags for {nominal, proper noun}+{verbal, possessive}, and existential *there* or predeterminer + verbal. This tagset had 20 tags, which proved manageable for a rapid short-term annotation effort. Other Twitter syntax projects, however, chose to adapt the PTB tagset, with the advantage that their data would be more closely compatible with existing resources and tools (Ritter et al., 2011; Foster et al., 2011a,b).

Twitter Treebanking. In annotating a treebank for Twitter, we estimated that a large volume of data at a coarse level of granularity would be more valuable for training parsers than a small amount of data with fine-grained labels. We thus developed Fragmentary Unlabeled Dependency Grammar (FUDG), an annotation scheme for unlabeled dependencies, and applied it to build the TWEEBANK corpus (Schneider et al., 2013; Kong et al., 2014). This scheme does make a couple of special distinctions—it provides special structures for coordination and multiword expressions, which occur in all text genres, and also allows multiple syntactic utterances/sentences per tweet—but by and large, it rests on the assumption that syntactic relations can be characterized as trees of head–modifier dependencies. (Accommodations for cases where it is difficult to determine those dependencies fully are described below.)

3 On Ambiguity

The third question asks: **Can the concept of "gold annotations" be applied to non-canonical languages where the inherent ambiguity in the data makes it hard to decide on the "ground truth" of an utterance?**

First, I think it is important to address the sources of ambiguity. The text that we encounter is (presumably) intended to be understood by someone. Of course, in unedited text there will be occasional errors—accidental misspellings, omitted words, etc.—that might render the utterance uninterpretable, and there may be fewer distinguishing orthographic cues (like capitalization). Even without production errors

[2] Unlike Petrov et al. (2011), we distinguished proper nouns from common nouns, as this distinction is beneficial for named entity recognition.

or orthographic ambiguities, the annotator may lack context that was available to the intended audience, or there may be genuine linguistic differences between the writer and annotator (e.g., unfamiliar slang). On occasion, we have to discard utterly uninterpretable utterances. In other cases we might misinterpret the utterance—but so long as it is a valid human interpetation, this is not necessarily a problem if the goal is to train a parser.

The FUDG framework (Schneider et al., 2013) provides a solution for some forms of syntactic ambiguity: it allows the annotator to **underspecify** parts of the parse. Essentially, the annotation provides a set of constraints which may be consistent with more than one tree. Tokens not mentioned in the constraints are unconstrained—they could be attached to any head in a full analysis consistent with the annotation.

It is also possible to constrain nodes' attachments without specifying their full structure. In Found the scarriest mystery door in my school . (shown with its annotation in the right side of figure 1), there is a subtle PP attachment ambiguity: what was in the school, the door or its discovery?[3] The annotation permits both possibilities via a **fudge expression**: the line `(Found* door in)` imposes the constraint that Found, door, and in must together form a connected subgraph, and (indicated by the asterisk) that Found must be the head of that subgraph. Thus, Found must have as daughters both door and in, or one of them, in which case the other one is the granddaughter to Found.[4]

4 The Annotation Process

When considering the merits of an annotation scheme, it can be easy to forget that the scheme will ultimately be embedded in an annotation process. A full **annotation framework** encapsulates the formal annotation scheme (e.g., tagset, units of annotation), linguistic

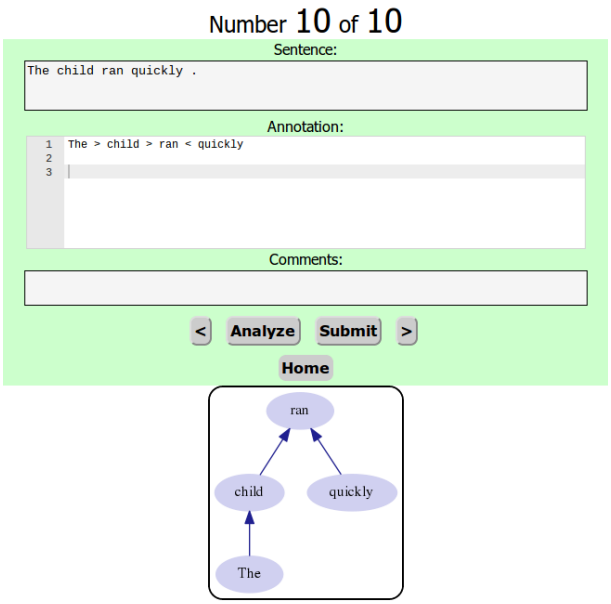

Figure 2: A simple training sentence in the FUDG/GFL annotation tool (Mordowanec et al., 2014).

conventions for its application, documentation, an annotation tool, a means of recruiting and compensating annotators, processes and materials for training annotators, procedures for validation and measuring inter-annotator agreement, etc. As suggested above, the design of the linguistic scheme cannot always be divorced from the practicalities of how it is to be applied to data. Likewise, not all tools and processes are appropriate for all schemes.

What are the considerations when choosing the level of expertise of the annotators? When is crowd sourcing appropriate? When do we need linguistic experts?

I find it useful to distinguish annotators along two dimensions. They can be **naïve**, **familiar**, or **expert** at understanding the linguistic phenomena of interest; and they can be **anonymous**—recruited from some general pool of users (such as Amazon Mechanical Turk), and possibly not serious about the task—or **trusted**—honestly willing to do what is asked of them (regardless of their *ability*). While there is crowdsourcing literature on making conventional annotation schemes more cost-effective with anonymous, naïve annotators (e.g., Snow et al., 2008; Hovy et al., 2014), success in this form of crowdsourcing requires the annotation task to be well understood (because it is more difficult to get useful feedback about challenging aspects of the task).

[3] Presumably both, semantically speaking. But this is not merely an issue of annotation conventions: if the scarriest mystery door in my school is a noun phrase, then the PP can be interpreted as expressing the set over which the superlative operates (i.e., 'the scariest out of all the doors in the school'); whereas if the superlative is functioning as an intensifier, it could be the scariest out of all doors in the world.

[4] I.e., `(Found* door in)` is consistent with any of the following: `Found < door < in`, `Found < in < door`, `Found < {door, in}`. The second of these, which is obviously incorrect, is ruled out by the first line of the annotation.

By contrast, the annotation schemes I have discussed above had never been piloted. We instead used a pool of local (trusted) annotators who were, for the most part, familiar with the fundamentals of POS/dependency representations but lacking in advanced training in syntax. Most of them were language technologies graduate students primarily trained as computer scientists. Given their fluency with text-based programming languages, we decided to formulate a similar language for FUDG dependency annotation—the Graph Fragment Language (GFL), whose notation is summarized in figure 1. In initial pilot studies, annotators were asked to annotate the data directly in text files, but this did not scale well because there was no immediate check for well-formedness of their input. Thus, for a larger annotation effort, we built a custom web interface for GFL annotations that produces an immediate graphical visualization of the parse (figure 2; Mordowanec et al., 2014). This framework seemed to work well, though we did not build a point-and-click treebanking interface for comparison.

Kong et al. (2014) present some analyses of the 900-tweet/12k-token TWEEBANK corpus. Most of its annotations were collected in a single a day from two dozen annotators, most of them *familiar* and a few of them *expert* with respect to syntactic representation and English grammar. Several quality measures are reported, but the main finding is that despite some noise in the data, training on TWEEBANK data (instead of out-of-domain training data) produces "a 7.8% gain [in parsing accuracy] with an order of magnitude less annotated data" (Kong et al., 2014, p. 1008). We take this as evidence that trusted non-expert annotations of linguistic structure can be useful. Whether naïve or anonymous annotators could be trained to do dependency annotation is an open question.

For building new resources for NCLs, is it still worthwhile to invest a huge amount of time and human labour for manual annotation, considering that the annotators spend most of their time making arbitrary decisions, and that the aim of building 'high-quality resources' for NCLs might not be realistic?

The Twitter syntactic annotation described above relied on fairly simple schemes distributed among many annotators over a short timeframe. The data produced by this approach has proved beneficial for training Twitter taggers and parsers—at least, relative to no in-domain data. The customization of the annotation schemes for the domain (including permitting underspecification) was intended to reduce the number of arbitrary decisions. (Our dependency annotation guidelines were fairly brief, and annotators were encouraged to avail themselves of underspecification when they encountered syntactic constructions not clearly addressed by the guidelines.)

It is, however, difficult to generalize beyond the framing of the tasks addressed here. I would not, for example, argue that the English Web Treebank (Bies et al., 2012)—a high-quality resource covering five genres of online text in the style of the Penn Treebank—was a wasted effort. But it will, I hope, permit experimentation testing whether the benefits of the full resource (for extrinsic tasks) can be approximated with smaller, less expert, cheaper annotations.

5 Why you shouldn't take my word for it

As with any annotation framework, it is difficult to say exactly which aspects of the setup were successful and which aspects could have been improved. To do so would have required a great many controlled annotation studies, whereas we were focused on producing as much useful data as possible given a limited budget. And of course, it's possible that a more conventional approach to annotation with fewer annotators would have produced more useful data.

In general, it has been my experience that—some well-established best practices notwithstanding—designing an annotation framework involves a mixture of guesswork, intuition, and trial and error. I hope future research will succeed at making this process more empirical and more predictable (see also Hovy and Lavid, 2010; Garrette and Baldridge, 2013). There is a great deal more to discover with regard to understanding the range of text varieties (Baldwin et al., 2013), building statistical models of annotator bias (Snow et al., 2008; Hovy et al., 2013; Passonneau and Carpenter, 2014), automatically detecting inconsistencies in linguistic data (Dickinson and Meurers, 2003; Loftsson, 2009; Kato and Matsubara, 2010), and bringing extrinsic models into the annotation loop (Baldridge and Osborne, 2004; Baldridge and Palmer, 2009; Settles, 2012).

Acknowledgments

I would like to thank the LAW organizers for hosting a panel on this topic, and Noah Smith, Bonnie Webber, and Archna Bhatia for their comments on a draft of this piece.

References

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In Dekang Lin and Dekai Wu, editors, *Proc. of EMNLP*, pages 9–16. Barcelona, Spain.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning actually work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proc. of EMNLP*, pages 296–305. Suntec, Singapore.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proc. of IJCNLP*, pages 356–364. Nagoya, Japan.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. Technical Report LDC2012T13, Linguistic Data Consortium, Philadelphia, PA. URL `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2012T13`.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proc. of EACL*, pages 107–114. Budapest, Hungary.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011a. #hardtoparse: POS tagging and parsing the Twitterverse. In *Proc. of the 2011 AAAI Workshop on Analyzing Microtext*, pages 20–25. San Francisco, CA.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011b. From news to comment: resources and benchmarks for parsing the language of Web 2.0. In *Proc. of IJCNLP*, pages 893–901. Chiang Mai, Thailand.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proc. of NAACL-HLT*, pages 138–147. Atlanta, Georgia, USA.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: annotation, features, and experiments. In *Proc. of ACL-HLT*, pages 42–47. Portland, Oregon, USA.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proc. of NAACL-HLT*, pages 1120–1130. Atlanta, Georgia, USA.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proc. of ACL*, pages 377–382. Baltimore, Maryland, USA.

Eduard Hovy and Julia Lavid. 2010. Towards a 'science' of corpus annotation: a new methodological challenge for corpus linguistics. *International journal of translation*, 22(1):13–36.

Yoshihide Kato and Shigeki Matsubara. 2010. Correcting errors in a treebank based on synchronous tree substitution grammar. In *Proc. of ACL*, pages 74–79. Uppsala, Sweden.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proc. of EMNLP*, pages 1001–1012. Doha, Qatar.

Hrafn Loftsson. 2009. Correcting a POS-tagged corpus using three complementary methods. In *Proc. of EACL*, pages 523–531. Athens, Greece.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Michael T. Mordowanec, Nathan Schneider, Chris Dyer, and Noah A. Smith. 2014. Simplified dependency annotations with GFL-Web. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 121–126. Baltimore, Maryland, USA.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-speech tagging for Twitter: word clusters and other advances. Technical Report CMU-ML-12-107, Carnegie Mellon University, Pittsburgh, Pennsylvania. URL `http://www.ark.cs.cmu.edu/TweetNLP/owoputi+etal.tr12.pdf`.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv:1104.2086 [cs]*. URL `http://arxiv.org/abs/1104.2086`, arXiv: 1104.2086.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: an experimental study. In *Proc. of EMNLP*, pages 1524–1534. Edinburgh, Scotland, UK.

Nathan Schneider, Brendan O'Connor, Naomi Saphra,

David Bamman, Manaal Faruqui, Noah A. Smith, Chris Dyer, and Jason Baldridge. 2013. A framework for (under)specifying dependency syntax without overloading annotators. In *Proc. of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 51–60. Sofia, Bulgaria.

Burr Settles. 2012. *Active Learning*. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, San Rafael, CA.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast — but is it good? Evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP*, pages 254–263. Honolulu, Hawaii.