Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–9

LogicalFormBanks, the Next Generation of Semantically Annotated Corpora: key issues in construction methodology

António Branco

University of Lisbon, Portugal

Abstract

The next generation of semantically annotated corpora will move a step further from raw text to meaning representation. The information to be encoded will go beyond the phrase-level information stored in PropBanks and represent sentence-level semantic information. In this paper I address issues that call to be explicitly articulated concerning the construction methodology of corpora annotated with logical forms for this construction to be practically viable.

Keywords: language resources, corpora, annotation methodology, LogicalFormBanks

1 Introduction[0]

An important methodological breakthrough took place in Natural Language Processing (NLP) with the advent of statistical approaches. This permitted important advances in terms of efficiency and robustness in tools of almost every area, ranging from tokenization to parsing, and in a wide range of applications, ranging from information extraction to machine translation.

These approaches are data-intensive and need large data sets for the estimation of relevant stochastic parameters as well as the subsequent evaluation of the corresponding classifiers. These data sets have steadily grown not only in terms of their size but also in terms of the complexity of the linguistic information they store, as the application of stochastic techniques have moved from relatively shallow (e.g. POS tagging) to more deep processing tasks (e.g. semantic role labeling).

Hence, development activities on annotated corpora have been deployed around extending lexical and morphological information with information concerning syntactic constituency (aka TreeBanks), syntactic functions (aka DependencyBanks), and most recently with phrase-level semantic functions and roles (aka PropBanks). The next generation of annotated corpora will expand these annotations with semantic information of different sorts beyond the phrase level, starting at the sentence-level representations of meaning (logical forms).

[0]This paper supports the invited plenary talk at the IIS'2009 conference by providing a broad overview of key issues concerning the annotation methodology of LogicalFormBanks. For the remaining issues addressed in this talk, please refer to the corresponding handout.

Based on the experience that at our team, we have been gathering in the construction of a LogicalFormBank.[1]

In this paper I aim at addressing generic issues worth being clarified concerning the construction methodology of this type of corpora for it to be practically viable. I seek to articulate into a single and focused outline ideas that if not in explicitly, at least implicitly or in part, have been circulating concerning the more adequate construction methodology for this type of corpora.

In Section 2, I present an overview that serves as the background for the matters being dealt with in this paper. The methodological issues on the construction of LogicalFormBanks are addressed in Section 3. Section 4 is devoted to elucidate a construction component that turns out to be specific of this type of corpora. This paper closes with concluding remarks in Section 5.

2 Annotated corpora: background

Language corpora are collections of representations of natural language utterances. They may include materials from a wide range of sources (e.g. written vs. oral), communicative situations (e.g. dialogue vs. letter), genres (e.g. novel vs newspaper), or domains (e.g. tourism vs. biotechnology). These collections have been extended with annotations by means of which there is an explicit association of their tokens with corresponding linguistic information, thus becoming not only archives of raw language materials but also key repositories for sharing linguistic knowledge.

Annotated corpora encode linguistic information from different dimensions in correlation with stretches of different spans (word, phrase, sentence, etc.). Most commonly, they bear morphological information about inflectional features and about POS. A great deal of effort has been put also into associating syntactic information concerning phrase constituents and categories, and/or grammatical function (Head, Subject, Object, etc.) in which case they tend to be termed specifically as TreeBanks (e.g. Marcus *et al.* 1993).

Recently, some corpora have been developed which are annotated with phrase-level semantic information concerning semantic functions (Specifier, Arg0, Modifier, etc.) and/or semantic roles (Agent, Theme, Beneficiary, etc.). Following Palmer *et al.* (2005), such annotated corpora have been termed as PropBanks.

As repositories of linguistic knowledge, annotated corpora have been key assets for natural language science and technology along at least the following lines:

- for quantitative and cross-language studies of natural languages;
- in the gathering of structured linguistic data such as lexicons, ontologies, etc.;
- in providing training data sets for NLP tools of different kinds, such as POS taggers, parsers, etc.;
- and in providing test sets for the evaluation of NLP tools and applications.

[1]For more information, see `http://semanticshare.di.fc.ul.pt`. I am grateful to Francisco Costa, Sara Silveira, Mariana Avelãs and Clara Pinto for their contribution for the maturing of the experience reported in this paper.

2.1 Past and recent generations of annotated corpora

As in the remaining areas of NLP, also here English is the best researched and equipped language. In terms of TreeBanks, the PennTreeBank, made of texts from the Wall Street Journal, is definitely a central reference (Marcus *et al.*, 1993). Not only for its size [1 M], only surpassed by the size of the Prague Dependency TreeBank [1.5 M] (Bohmova *et al.*, 2003), but above all because it became the de facto standard repository of linguistic knowledge upon which in the last decade most of the research on stochastic-based NLP was carried on. In terms of TreeBanks, it is also worth mentioning the TIGER project for the German language (Brants *et al.*, 2002), for its continuity, and for the documentation services and tools that it has made available for the community.

The TIGER site lists TreeBanks developed or in development (manually annotated/verified) for Czeck [1.5M], Basque, Bulgarian, Danish [100K], French [650K], German [900K], Hindi, Italian, Japanese, Slovene [30K], English [1M], Dutch [150K], Polish, Spanish and Turkish.

In the more specific area of PropBanks, the key reference is the English PropBank (Palmer *et al.*, 2005). This is the first large scale corpus with annotation for semantic roles [1M]. Similar PropBanks have been developed so far for Chinese [250K] and Korean [200K].

In the area of corpora with sentence level grammatical annotations, including semantic annotations, for their ground breaking results, it is worth pointing out the works by Dipper (2000) and Mullen *et al.* (2001) and the HPSG-based Redwoods, Hinoki and Trepil corpora (Oepen *et al.*, 2002; Fujita *et al.*, 2006; Trepil, 2006).

2.2 Major research issues

The development of annotated corpora benefits from the lessons learned and the tools developed in the scope of the simpler sub-tasks of compiling, archiving and searching raw corpora. It presents however specific challenges that result from the need to handle the complex information being annotated.

As for archiving, new challenges had to be faced in terms of selecting appropriate encoding schemes and formats compliant with industrial demands and standards (TEI, 2004; XCES, 2006; LMF, 2009), and of defining suitable tools to provide for the management of the corpora and the conversion between different formats (e.g. Ide and Brew 2000).

From the point of view of the annotation work proper, a bunch of issues need to be properly addressed:

- methodologies to improve insensitiveness to annotation errors and differences in the application of annotation guidelines by the human annotators (e.g. Palmer *et al.* 2005);
- determination of upper-bounds for correct annotation by humans and measuring inter-annotators agreement (e.g. Artstein and Poesio 2009);
- design of efficient annotation techniques and tools to reduce the effort and time required for manual annotation/verification (e.g. Plaehn and Brants 2000);

- development of tools for automation of corpus validation and consistency checking across layers (e.g. Cotton and Bird 2002).

Finally, the annotated corpora offer new challenges in terms of ensuring the access to highly structured data, for which specific search and visualization tools have to be prepared (e.g. Keller *et al.* 1999).

2.3 Next generation of semantically annotated corpora: LogicalFormBanks

The next generation of semantically annotated corpora will move a step further from raw text to meaning representation. The information to be encoded will go beyond the phrase-level semantic information stored in PropBanks and include the semantic representation (logical form) of the sentence.

The linguistic information to be encoded in these LogicalFormBanks will reach a new level of sophistication where the utilization of annotation tools is no longer a matter of convenience but of necessity. They have to integrate auxiliary grammars and lexicons for deep linguistic processing (Bos and Delmonte, 2008) in order:

- to obtain deep, accurate sentence-level semantic representations (logical forms) to serve as annotation materials, which can be selected from parse results but cannot be massively and accurately drawn or corrected by hand;
- to bring into the annotation process the benefits of principled linguistic theorizing of a more deep level than the shallow ones that has been put to use in the construction of previous generations of TreeBanks and PropBanks;
- to ensure the correct alignment and integration of annotations pertaining to the different linguistic dimensions and layers.

3 Construction methodology

As expected, the construction of LogicalFormForms raises a range of challenges along the dimensions and research issues mentioned above.

Extension of encoding schemes and formats, towards de facto standards, have to be found; the annotation guidelines turn out here to be the whole grammar specification, and new procedures may have to be devised to minimize annotation errors and differences in the application of the "guidelines"; known inter-annotator agreement indicators will be adapted or new ones may be needed; new search and visualization tools will be developed, etc.

It is not within the scope of the present paper to address all such aspects. Here I will focus on the basic topic of the construction methodology of annotated corpora itself and on key issues that in this regard LogicalFormBanks bring about.

3.1 TAVA: Train, Annotate, Validate, Adjudicate

The experience gathered in the development of annotated corpora in the last decades permitted to build a consensus on the basics of the most convenient construction methodology. The annotation work progresses by the successive execution of a typical four step procedure, which includes:

- *train*: on the basis of the material whose annotation was concluded so far, train a classifier that automatically assigns annotation "tags";
- *annotate*: run this classifier on the material whose annotation is yet to be done;
- *validate*: let the human annotators validate the result of the classifier by either confirming or correcting its output annotation;
- *adjudicate*: submit the materials validated by different annotators to a final supervision step by a human adjudicator who settles possible divergences among the different annotators.

This TAVA methodology incorporates measures to minimize annotation errors (by means of multiple validations followed and harmonized by adjudication). Importantly, it incorporates also measures to minimize the time and effort spent in the development of the corpus. No matter how suboptimal the annotation tool may be, its usage accelerates the construction of the corpus. It takes less time and effort to correct only part of the items to be annotated—those where the tool failed—than to manually annotate every item from scratch.

In case the auxiliary annotation tool is based on a classifier whose performance improves with larger training data, then the TAVA procedure should be iterated along the construction timeline (yielding successively improved versions of the tool) to further minimize annotation time and effort.

3.2 From linear to circular TAVA

Under this methodology there is thus a parallel and continuous progression in terms of timeline of the construction process, performance of auxiliary annotation tools, and corpus coverage with accurate annotation. At each iteration of the TAVA procedure, the auxiliary annotation tool typically tends to improve its performance in the assignment of a tag out of the predefined tagset.

When applying this methodology to the construction of LogicalFormBanks, the auxiliary annotation tool is now the grammar whose output are grammatical representations, that include logical forms. As expected, progression in terms of the development of the grammar and its performance may involve reestimating stochastic parameters to resolve parsing ambiguity, as larger portions of the annotated corpus become adjudicated.

As the construction of the corpus progresses, the annotation time and effort is minimized as on average, per sentence to be annotated, there will be less "annotation tags" (less parse results) to be visited by the human annotators until the correct parse is found in the ranked list of parses. At each iteration of the four step procedure, there will be more training data available, which supports the training of more accurate parsing disambiguators. Consequently, for each sentence, the ranked list of its possible parses produced by the grammar will more likely display the appropriate parse—which it should be annotated with—closer to the top of the list.

But, crucially, progression in terms of the development of the grammar and its performance involves also extending the empirical coverage of the grammar so that new linguistic constructions may be handled by it. This implies that sentences in the corpus that in the previous passes were not annotated—did not receive a

parse because they contain linguistic phenomena not handled yet at that time by the grammar—, may become annotated with a new version of the grammar. Accordingly, instead of continuous coverage, we should expect here interpolated coverage for the annotated corpus.

Furthermore, for certain linguistic constructions already within the grammar empirical coverage, the depth and sophistication of the grammatical analysis produced may be revised or upgraded as the development of the grammar proceeds. This implies that sentences in the corpus which in the previous passes were already annotated by the grammar—and whose annotation may have been already validated and adjudicated—, may become now obsolete with a new version of the grammar, and have to be extended or even reannotated.

In this connection, it is important to notice two important impacts in what concerns the construction of LogicalFormBanks methodology.

First, instead of a fixed set of annotation "tags", one should be ready for a tagset that evolves both in terms of its size and in terms of the "tag" themselves.

Second, the usual linear progression of the annotation effort does not hold for LogicalFormBanks. For the TAVA methodology to be applied, the corpus to be annotated should be conceived here not as a double ended string of items to be annotated, but as a circular one.

Circular TAVA methodology permits that during the development of the corpus, a given sentence may receive more than one pass of the grammar. This ensures that if the grammar at some stage of its progression starts producing a grammatical representation for that sentence, the latter will eventually gets annotated with that representation in some subsequent pass.

Circular TAVA ensures also that if due to the evolution of the grammar, the grammatical representation a sentence was previously annotated which needs to be reviewed or replaced, the human annotators will be offered the chance to do it. In some subsequent pass of the grammar, this sentence will be taken care of as it will stand alongside with the sentences with no annotation yet.[2]

It is worth noting that this does not impose that for every new version of the grammar the whole corpus has to be re-passed through. To ensure consistency, only the final pass, with the last version of the grammar, has to perform a complete scan of the corpus.

It is tempting to consider abandoning the TAVA methodology for the construction of LogicalFormBanks and fall back to a simplifying approach where only one pass of the corpus to be annotated would be executed, with the very final version of the grammar. This however is not a practically viable alternative. This would cut the crucial feedback for the construction of the grammar coming from its application, and partial failure, over the corpus to be annotated. And this would ignore also that, due their very nature, there is no known grammar of this type for any language that has reached its final, complete version.

[2]Note that this is less penalizing than what it may seem: when a new version of the grammar produces a new set of grammatical representations for a sentence which was already annotated, and one representation in that new set unifies with the previously annotated representation, the old representation can be replaced by the new one without human reannotation being needed. For example, [incr tsdb()] is a grammar development tool that supports this methodology (Oepen, 1999).

4 A key construction component: the RCC Corpus

To a large extent, the above demands can be seen as natural extensions, to the new type of semantically annotated corpora, of key components of the construction methodology matured in the construction of previous types of annotated corpora. I would like to focus now on what may be seen as a component that is specific of the construction of LogicalFormBanks.

It may happen that with a new, more advanced version of the grammar, a sentence that was previously annotated, receives no annotation. For the sake of concreteness, let us consider the following example. In version n, the grammar was not prepared to handle yet, say, negative concord. Sentences with negative words such as *not*, *nobody* or *never*, etc., were required only that these words were classified, say, as negation adverbs. In version n+1, the grammar evolved to accommodate negative concord, though yet under an incomplete account. In order to capture key aspects of the negative concord, the cooccurrence of negative words were duly constrained. But as this was yet an incomplete account, to be extended in subsequent grammar versions, only nominal quantifiers are considered, and sentences with temporal items in negative concord, like *never*, receive no parse.

Accordingly, a sentence with negative concord involving temporal quantifiers that under version n of the grammar was annotated with a grammatical representation, under version n+1 receives no representation. In order to keep the consistency among the annotations in the sentences of the corpus, and between the annotated corpus and the grammar, the previous annotation is abandoned and, for now, that sentence remains unannotated.

This example helps to understand that during the construction process of the annotated corpus, a sentence may lose its previous annotation. There is no big deal here: it will be annotated again, with a new and appropriate grammatical representation later on with a more mature grammar version.

In this connection, the crucial point worth noting is that it may happen that a sentence that should not lose its annotation actually loses it. This may occur when the development of some part of the grammar – to account for some linguistic phenomenon – has undesired side effects on the account of other orthogonal phenomena. This is a well known situation for grammar developers, that tends to occur in the construction of grammars that grow to a point where they become very large in terms of empirical coverage and range of linguistic phenomena dealt with.

Now, the critical issue is that these two types of situations need to be distinguished one from another, that is the acceptable from the undesired loss of previously annotated representations. By not controlling for undesired loss of previous annotation, one runs into the risk of entering into a corpus construction process without progress or even with regression in terms of corpus coverage.

Note that the distinction between the two types of losses is something that cannot be done on the fly by the human annotators that arrive at a sentence that lost its annotation from version n to version n+1. In large scale development, in order to optimize resources, the group of grammar developers and the group of annotators are disjoint, and the later would not be prepared to control for that distinction, which in many cases may not be trivial even for skilled grammar

developers.

Therefore, a more principled approach is called for here. It can be based, alongside the primary corpus to be annotated, on a companion annotated corpus that supports the task of controlling the progression of the primary corpus construction process. This regression control companion (RCC) corpus collects hand made sentences, together with their accurate annotations, that the grammar cannot not parse.[3] Every time a new version of the grammar enters into use for the annotation of the primary corpus to be continued, it will be run first over this corpus. This offers the chance for human annotators to make sure that no previously annotated sentence in this corpus changed its status to unannotated.

Being instrumental in the construction of the primary corpus to support empirical NLP research, ideological prejudice should not lead us to dismiss this companion corpus on the basis that it includes hand made sentences.

5 Concluding remarks

In this paper, I sought to articulate into a single and focused outline key demands for the annotation methodology of LogicalFormBanks.

Given the sophistication of the linguistic information to be associated with natural language expressions and the formal complexity of its representation, the usage of an annotation of tool is any longer a matter of convenience but of necessity here. Such annotation tools are deep linguistic processing grammars delivering grammatical representations that include a logical form representation of the input sentence.

Given the nature of these grammars and of their development process, it turns out that the annotations and the "tagset" should be expected to evolve alongside with the construction of the annotated corpus. This is at the root of a crucial change from a linear to a circular Train-Annotate-Validate-Adjudicate procedure underlying the annotated corpus construction methodology.

It turns out also that, in contrast with previous types of annotated corpora, the construction of LogicalFormBanks requires a specific component, a secondary data set, under the form of a regression control companion corpus.

References

Artstein and Poesio (2009), Inter-Coder Agreement for Computational Linguistics, *Computational Linguistics*, 35:555–596.

Bohmova, Hajic, Hojicova, and Hladka (2003), The Prague Dependency Treebank: The-Level annotation scenario, in Abeillé, editor, *Treebanks*, Kluwer.

Bos and Delmonte, editors (2008), *Semantics in Text Processing*, College Publications.

Brants, Dipper, Hansen, Lezius, and Smith (2002), The TIGER Treebank, in *Proceedings of TLT02*.

[3] For each linguistic construction the grammar coverage is extended with, this secondary corpus should be extended as well with a list of corresponding annotated sentences. For methodological issues concerning grammar test suites, see (Oepen *et al.*, 1997).

Cotton and Bird (2002), An Integrated Framework for Treebanks and Multilayer Annotations, in *LREC2002*.

Dipper (2000), Grammar-based Corpus Annotation, in *Workshop on liguistically interpreted corpora*, pp. 56–64.

Fujita, Tanaka, Bond, and Nakaiwa (2006), An implemented Description of Japanese: The Lexeed dictionary and the Hinoki treebank, in *COLING/ACL 2006*.

Ide and Brew (2000), Requirements, Tools, and Architectures for Annotated Corpora, in *Proceedings of Data Architectures and Software Support for Large Corpora*, Paris.

Keller, Corley, Corley, Crocker, and Trewin (1999), Gsearch: A tool for syntactic investigation of unparsed Corpora, in *LINC 1999*.

LMF (2009), Language Resource Management – Linguistic Annotation Framework, iSO/TC37/SC4 `http://www.iso.org/iso/iso_catalogue/catalogue_tc`.

Marcus, Santorini, and Marcinkiewicz (1993), Building a large annotated corpus of English: The Penn Treebank, *Computational Linguistics*, 19(2):313–330.

Mullen, Malouf, and Noord (2001), Statistical Parsing of Dutch using Maximum Entropy Models with Feature Merging, in *Proceedings of the NLP Pacific Rim Symposium*.

Oepen (1999), [incr tsdb()] — Competence and Performance Laboratory. User Manual, Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany.

Oepen, Netter, and Klein (1997), TSNLP – Test suites for Natural Language Processing, *Linguistic Databases, CSLI Lecture Notes*, 77:3–28.

Oepen, Toutanova, Shieber, Manning, Flickinger, and Brants (2002), The LinGO Redwoods Treebank: Motivation and Preliminary Applications, in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 1253–7, Taipei, Taiwan.

Palmer, Gildea, and Kingsbury (2005), The Proposition Bank: A Corpus Annotated with Semantic Roles, *Computational Linguistics*, 31(1).

Plaehn and Brants (2000), Annotate—An Efficient Interactive Annotation Tool, in *ANLP2000*.

TEI (2004), Guidelines for Electronic Text Encoding and Interchange, Text Encoding Initiative, `http://www.tei-c.org`.

Trepil (2006), `http://gandalf.aksis.uib.no/trepil`.

XCES (2006), Corpus Encoding Standard for XML, `http://www.cs.vassar.edu/XCES`.