excellently summarizes the research tradition in the field of historical syntax. The chapters include Modern English syntax, Old English, Middle and Early Modern English, and Modern English since 1700. While this research manifests a Chomskyan influence, it is also supplemented by performative analysis and case grammar.

Her other research contributions focus on language change, particularly grammaticalization, lexicalization of conversational inferences, and subjectification; clause combining, particularly conditionals, concessives, and causal clauses. Some of her important articles in these areas include '*And* and *But* connectives in English' (*Studies in language*, 1986), 'On the rise of epistemic meanings in English: an example of subjectification in semantic change' (*Language*, 1989). With respect to clause combining, the investigation of the historical paths of English connectives *and* and *but* shows that there is a strong tendency for spatial and temporal meanings to give rise to logical connective meanings. In regard to the subjectification process in semantic change, it is demonstrated that "meanings tend to become increasingly situated in the speaker's subjective belief state or attitude towards the proposition." The semantic changes of modal auxiliaries (e.g., *must*), assertive speech act verbs, and modal adverbs show that epistemics develop from more root, deontic, and concrete meanings to more strongly subject epistemicity.

Her seminal book *Grammaticalization* (1993; 2nd edn. 2003, with P. Hopper) deals with historical as well as synchronic linguistics whereby "lexical items and constructions come in certain linguistic contents to serve grammatical functions, and, once grammaticalized, continue to develop new grammatical functions" (2003: xv). This change corresponds to the shift of content words to pure function words.

Traugott also co-authored *Regularity in semantic change* with R. Dasher (2001) in which cross-linguistic unidirectional tendencies in semantic change are discussed in detail. Her writings also include *Linguistics for Students of literature* (1980; with M. L. Pratt), and *On conditionals* (1986; co-edited).

See also: Grammaticalization.

Bibliography

Hopper P J & Traugott E (2003). *Grammaticalization* (2nd edn.). Cambridge: Cambridge University Press.

Traugott E (1972). *A history of English syntax: a transformational approach to the history of English sentence structure*. New York: Holt, Rinehart and Winston.

Traugott E & Dasher R (2001). *Regularity in semantic change*. Cambridge: Cambridge University Press.

Traugott E & Heine B (eds.) (1991). *Approaches to grammaticalization*, 2 vols. Amsterdam: Benjamins.

Treebanks and Tagsets

J Hajič, Charles University, Prague, Czech Republic

Introduction

Treebanks, by definition, are series of trees that arise from parsing a textual form of a sentence (as it appears on a printed page or in an electronic article, or that is a transcribed spoken utterance). Parsing of a sentence is an intermediate step on the way from a surface expression (written or spoken) to a formal sentence understanding representation. It is believed that parsing is a crucial step on that way, being the first one (after tokenization or acoustic processing, phonological, morphological, and other 'low-level' procedures) that assigns a structure to the analyzed utterance. Treebanks thus capture the result of such parsing of many utterances (typically, a continuous series of sentences and documents originally collected in an electronic corpus).

Treebanks are created (annotated) manually, by linguists (or by non-linguists with some linguistic training and under supervision by linguists), who also prepare and gradually refine the treebank annotation guidelines. The typical size of a treebank is hundreds of thousands to one or two million word tokens (tens to hundreds of thousands of utterances).

The purpose of treebanks is essentially twofold. First, they serve as an exact, explicit formal description of language structure under a given theory or formalism for further linguistic research (whether structural, lexical, or any other). Second, they can be used for automatic training of parsers – by machine learning, probabilistic, statistical, or other data-driven algorithms to derive features, rules, or other formal mechanisms (possibly with probabilities, weights, or scores) that can be then used to automatically replicate on some other, previously unseen data, the process of parsing that leads to the original treebank. Using such a trained parser, a much larger

treebank can be constructed automatically, even though only the best automatic parsers achieve less than 10% errors (for a discussion of error rate, see the section 'Using Treebanks: Searching and Parsing' in this article). Although automatically parsed treebanks can serve well for linguistic research that can tolerate the parsing output noise, they cannot be relied upon by further automatic processing as can the original, manually annotated treebanks, and they can be used (with caution) only for certain specific purposes.

Manual treebank annotation is a lengthy, expensive process (both in terms of human resources and actual money spent during the process). It is expected that treebanks will be reused for many projects for both of the aforementioned purposes; they are typically published for worldwide distribution by one of the leading publishers of electronic language resources, the Linguistic Data Consortium (Philadelphia, PA, USA) or ELRA/ELDA (Paris, France).

Types of Treebanks

Known treebanks can be classified into two groups according to the underlying view of the linguistic theory used:

- parse-tree (constituent, phrase-structure) treebanks, and
- dependency treebanks

Parse-tree treebanks can be almost, without change, viewed as derivation trees of some context-free grammar (CFG), with the original theory of phrase-structure grammars (PSG) being closest in its formulation. Dependency treebanks are typically close to the functional view, such as the Functional Generative Description Theory or the Meaning-Text Theory.

Parse-Tree Treebanks

A context-free grammar backbone is behind all such treebanks. Each node has in principle only a single label. The root of the tree (typically denoted S, for 'Sentence', or 'Start symbol' in CFG terminology) is typically split to two or more subtrees with other phrase labels (such as NP for noun phrase, VP for verb phrase, etc.; these are called non-terminal symbols in CFG terminology). Recursively, the subtrees are split further and further, until a terminal node (a node with no descendants) is reached; the terminal nodes correspond to the surface form of words in the analyzed sentence. The nodes just above the terminal nodes are called pre-terminals; they are labeled not by phrase labels, but by the POS (part-of-speech) tags assigned to the words at the terminal nodes below them. The number of nodes in a parse-tree treebank (if pre-terminal nodes are used) is thus between twice and three times the number of tokens in the analyzed sentence, or even more if some non-branching non-terminals are employed in the tree. An example of a tagset used for a parse-tree treebank follows.

```
(S (NP Industrial production)
   (VP increased
       (PP by
           (NP 0.8 percent))
       (PP in
           (NP December))))
```

Figure 1 Example parse tree with phrase labels *(Industrial production increased by 0.8 percent in December)*.

S (Sentence)
NP (Noun Phrase)
VP (Verb Phrase)
AUXV (Auxiliary Verb)
ADJP (Adjectival Phrase)
ADVP (Adverbial Phrase)
PP (Prepositional Phrase)
SBAR (Relative Clause, Subordinate Clause)
FRAG (Fragment)
PRN (Parenthetical)
CONJP (Multi-word Conjunction)
UCP (Unlike Coordinated Phrase)

Figure 1 shows an example of an annotated parse-tree.

Dependency Treebanks

In dependency treebanks, no non-terminal or phrase-marking labels are used. Instead, nodes in the dependency tree are labeled by complex labels (attribute-value pairs). There are at least three of them at each node: the word (in its surface form) itself, its POS tag, and some sort of a functional label. The edges of the tree denote dependency: from the root to the leaves, the 'upper' node of an edge is called the 'governor' and the lower one is called the 'dependent.' Even though the direction of the dependency is sometimes theory-dependent, verbs typically govern its complementations (arguments and adjunct), nouns their attributes and other modifiers, conjunctions the subordinate clauses they precede, prepositions the head noun of the phrase they belong to, etc. The number of nodes in a dependency treebank is thus equal to the number of tokens in the analyzed sentence.

A possible list of dependency functional labels in such a treebank follows.

Pred (Predicate)
Sb (Subject)
Obj (Object)

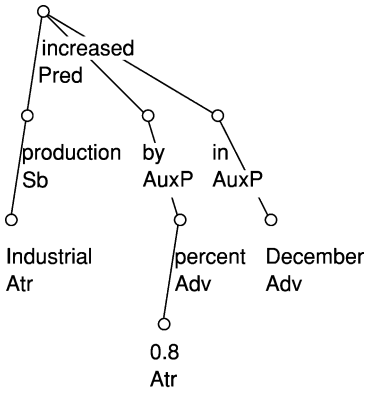

Figure 2 Example surface-syntactic dependency tree *(Industrial production increased by 0.8 percent in December).*

Adv (Adverbial (all types))
Pnom (Nominal part of a predicate)
Atr (Attribute)
AuxV (Auxiliary verb (to_be))
AuxP (Preposition)
AuxC (Subordinate conjunction)
AuxZ (Intensifier)
AuxG (Graphical symbol)
AuxR (Reflexive particle)
Coord (Coordination)

Figure 2 shows their usage in a partially annotated syntactic dependency tree.

Type Difference

Treebank types are, after all, not as different as they might seem from their description. Parse-tree treebanks can be converted into dependency treebanks if heads (governing nodes in the resulting converted dependency tree) are determined for every non-terminal in the corpus, which might be a very non-trivial task if they are not manually annotated. Dependency treebanks can be converted into parse-tree treebanks under two basic conditions: first, a strategy has to be determined for whether the resulting trees should be primarily deep or wide (or something in between, based on the language, style of annotation, purpose of conversion, etc.), and second, a procedure for naming the newly created non-terminal nodes must be devised that can use only the existing dependency type labels. Moreover, if the dependency trees are not projective, additional considerations must be taken into account.

While the parse-tree style of annotation can be (almost) directly used for training automatic parsers, the dependency trees are believed to have the advantage of being closer to semantic interpretation of the sentence and therefore more suitable for further (deeper) analysis.

The Complexity and Depth of Annotation: From Syntax to Semantics and Beyond

The basic types of treebanks already described are also referred to as (surface) 'syntactic treebanks.' Various schemes have been designed to 'deepen' the annotation towards a (syntactico-)semantic, pragmatic, content or 'logical' annotation. For example, the predicate-argument structure of verbal or nominal clauses is annotated (similarly in treebanks based on other theories, valency structure of verbs and some nouns and adjectives can be annotated), function words (and/or inflection for languages that use it extensively) can be generalized and formalized into a detailed set of node function labels, and various grammatical functions can be reflected in an enriched set of node labels. More structure that complements the basic tree structure of the sentence analysis representation is introduced for anaphora links (and co-reference in general), and the information and discourse structure of the sentence is annotated as well. These various schemes always contain links back to the basic syntactic annotation or are directly embedded in it in order to facilitate automatic learning of the process of 'deepening' the sentence analysis (and vice versa). Examples of functional (dependency) labels follow.

PRED (Predicate)
ACT (Actor)
PAT (Patient (2nd verbal argument))
ADDR (Addressee)
EFF (Effect)
BEN (Benefactor)
LOC (Location)
TWHEN (Time-when)
THO (Time-how often)
AIM (Aim)
CAUS (Cause)
RSTR (Restrictive Attribute)
DIFF (Difference)
DISJ (Disjunctive coordination)
RHEM (Rhematizer)

Figure 3 shows an example of a deep syntactico-semantic annotation.

Lexicons and Treebanks

Treebanks are sometimes accompanied by lexicons that contain the context-independent information about lexical behavior of words at the given level of annotation. For inflectionally rich or agglutinative languages, a morphological dictionary is often provided that lists many more words than can be found in the annotated treebank, facilitating processing of

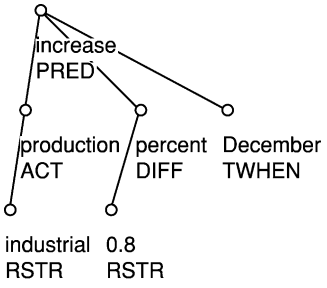

Figure 3 Example deep syntactic (functional) sentence representation *(Industrial production increased by 0.8 percent in December).*

general texts. For deeply (semantically) annotated corpora, a 'valency' lexicon or a 'PropBank' lexicon (both essentially listing possible arguments of words of various parts of speech, in a theory/annotation dependent way) accompany such a corpus, often with links between the lexicon entry occurrences in the corpus and the lexicon entries themselves, for easy reference.

Parallel Treebanks

For multilingual applications, most notably for machine translation, parallel treebanks provide an advantage to ordinary parallel corpora in that they can be used for inducing reliable structural correspondence between the languages in question. It is believed that structural differences between languages, especially at the semantic level, are much smaller than the differences observed at the word level (such as different word order, use of function words vs. inflection in the other language, etc.). Very few parallel treebanks exist today, but some are under construction.

The Process of Treebank Annotation

Manual annotation of a million-word (or bigger) treebank is a complex task that includes several stages and a varying size of the annotation team. The annotation guidelines have to be written first (even though they are necessarily being modified and extended during the process). Data has to be selected and technologically prepared for annotation, and annotation software has to be acquired or created. Decisions have to be made about how much automatic pre-annotation is desirable given the main goals of the annotation project and the time and budget constraints; one has to be cautious here because the team members may be influenced by the pre-annotation. Annotators have to be chosen and trained, and their ability to follow the guidelines, work effectively, and focus for many hours (which is often unrelated to their education or age) has to be assessed before the final selection is made.

During the process of annotation, inter-annotator agreement has to be measured and problems must be solved quickly by modifying the guidelines and/or retraining the annotators. Software has to be maintained and enhanced during the process of annotation to make the process more efficient and less error-prone. If the teams of annotators are specialized (to annotate only a subset of the node labels, for example) and work in parallel on the same data, extreme care has to be taken when merging the annotation into a single final corpus. Access to dictionaries and the annotated corpus (if the annotation is performed onsite and/or online, with direct access to the actual data) must be controlled in order to avoid simultaneous changes, much as in large online databases. When the annotation is finished, it has to be extensively checked at various levels (low-level markup, consistency with the final version of the guidelines, consistency between the groups of annotators; consistency inside the data, if possible; link targets for additional superimposed structure; etc.).

Last, the result must be prepared for publication with full documentation, examples of use, viewing and processing software for general use, pointers to papers and articles about the underlying theory, current use, or other aspects of annotation, pointers to similar or complementary projects, etc.

Typically, the annotation itself as performed by the annotators forms only 15 to 30% of the total effort. The cost of producing a syntactic treebank of a reasonable size is about $1 per annotated word, and can triple (or worse) if some form of a complex, 'deep' semantic annotation is used.

Existing Treebanks: a Short Survey

The first and most used treebank of all time is the Penn Treebank (Marcus *et al.*, 1993). It is of the parse-tree type and contains 1.3 million annotated words (English articles from the Wall Street Journal from the end of the 1980s and beginning of the 1990s.) The treebank has been enhanced since its first publication (e.g., the set of non-terminal labels has been extended to contain functional information as well). Recently, predicate-argument annotation has been superimposed on the Penn Treebank, based on the PropBank (Kingsbury and Palmer, 2002) and NomBank (Meyers *et al.*, 2004) projects.

The Czech-language Prague Dependency Treebank project (Hajič *et al.*, 2001) uses the dependency-type of annotation (based on the Functional Generative Description theory (Sgall *et al.*, 1986)), and consists of two treebanks whose nodes are linked together:

one contains the annotation of the morphological and surface syntactic structure of Czech, and the other one (on the same data) contains the 'deep' complex annotation of valency, detailed dependency and grammatical functions (a total of 16 node labels), co-reference, and information structure of the sentence.

The German TIGER (Brants *et al.*, 2002) corpus (an extension of an earlier and smaller Negra corpus) contains a mixture of both types of annotation (parse-tree and functional/dependency). On top of the usual non-terminals used for denoting the underlying parse-trees, a wide-coverage LFG grammar formalism has been used for the functional style of annotation. Verb subcategorization is also used as part of the annotation. The corpus has been prepared semi-automatically; however, this corpus is considered a manually annotated one because of the amount of manual work and checking that went into its creation.

There are other treebanks being used and/or created, such as for the Arabic, Bulgarian, Chinese, Danish, Dutch, Hungarian, Italian, Slovak, Slovenian, Swedish, Turkish, and other languages. These treebanks are typically smaller in size, and they are modeled after one of the basic types; sometimes (for example, the Arabic one) they are using (almost) the same sets of labels as the Penn or the Prague Dependency Treebanks. The largest spoken language treebank is the English Switchboard corpus (Godfrey *et al.*, 1992); however, it has been parsed automatically (not manually) by one of the statistical parsers trained on the Penn Treebank.

Using Treebanks: Searching and Parsing

Linguists are interested in searching treebanks, which is a much more complex task than searching an ordinary (linearly ordered) corpus, both from a mathematical and an end-user point of view. Instead of – or in addition to – formulating corpus queries in terms of distance between words, syntactic relations are the core of the queries. For example, one might search for all clauses in which two direct objects are used, regardless of the verb that heads such a phrase. In the case of deeper annotation, one might formulate even more complex queries, such as finding all instances of personal pronouns that refer outside a clause boundary (and perhaps to express their count as the percentage of all personal pronouns used in the corpus). Currently, very few general-purpose tree search systems exist. A rudimentary search system is called `tgrep`, a tree search system operating on principles similar to those of the file-searching `grep` utility (and suitable especially for the Penn Treebank), but without a graphical user interface. TIGERSearch, a tool to search corpora following the TIGER treebank style of annotation, does contain such an interface, as does the Netgraph system that accompanies some of the treebanks published by the Linguistic Data Consortium. Other systems are currently being developed, usually to accompany treebanks developed by various research groups.

The Computational Linguistics community sees the main purpose of treebanks in automatically training parsers. The manually annotated data is fed into a system that automatically derives features or rules and/or their weights or probabilities from the manual annotation and stores them as a parsing 'model'. Once trained, such parsing model used with appropriate parsing software can automatically replicate the manual annotation on new texts and thus be eventually made part of natural language processing applications. Good parsers, however, do not have to necessarily be trained automatically (before the advent of treebanks, it was not even possible to do so). The logical question then arises when comparing parsers of any kind – which of them is better? Again, treebanks provide the answer: parser output can be compared to the manually annotated data (obviously, it is different than that used for training a data-driven parser) and numerically evaluated. The error function used depends on the type of treebank. In the case of parse-tree treebanks, the measure concerns the number of 'crossing brackets (and their labels)', i.e., roughly speaking, the number of subtrees whose spans do not match that of the manually annotated sentence. For dependency treebanks, the measure is even simpler – every node that is not correctly attached to its 'true' (i.e., manually annotated) governor, or has the wrong label, counts as an error.

Summary

Treebanks, since the publication of the first one, the Penn Treebank, have been the source of a true revolution in Computational Linguistics, especially in the area of parsing natural language sentences. With limited research resources and the apparent high cost of building treebanks, we cannot currently hope for substantially larger manually annotated treebanks – despite the desperate need for them; further research is needed to discover ways of annotating more textual and spoken data with the reliability attainable by humans but using much more efficient methods.

See also: Anaphora and Coreference Resolution, Statistical; Computational Lexicons and Dictionaries; Corpora; Corpus Linguistics; Dependency Grammar; Language

Processing: Statistical Methods; Mark-up Languages: Text; Parsing: Statistical Methods; Part-of-Speech Tagging.

Bibliography

Abeillé A (ed.) (2003). *Treebanks: building and using parsed corpora* (vol. 20). Series: Text, Speech and Language Technology: Kluwer Academic Publishers.

Bird S & Liberman M (2001). 'A formal framework for linguistic annotation.' *Speech Communication 33(1,2),* 23–60.

Brants S, Dipper S, Hansen S, Lezius W & Smith G (2002). 'The TIGER Treebank.' In *Proceedings of the first international workshop on treebanks and linguistic theories.* Sozopol, Bulgaria. 24–42.

Charniak E (1996). *Statistical language learning.* Cambridge, MA: MIT Press.

Cotton S & Bird S (2002). 'An integrated framework for treebanks and multilayer annotations.' In *Proceedings of the third conference on language resources and evaluation (LREC 2002).* Las Palmas, Spain. European Language Resources Association (ELRA), France. 1670–1677.

Garside R, Leech G & McEnery T (1997). *Corpus annotation: linguistic information from computer text corpora.* Harlow: Addison Wesley Longman.

Godfrey J J, Holliman E C & McDaniel J (1992). 'Switchboard: a telephone speech corpus for research and development.' In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing, vol. I.* Institute of Electrical and Electronics Engineers (IEEE), USA. 517–520.

Hajič J *et al.* (2001). *The Prague dependency treebank* (version 1.0). Linguistic Data Consortium, Philadelphia, PA, USA. CD-ROM. Catalog no. LDC2001T10.

Jelinek F (1998). *Statistical methods for speech recognition.* Cambridge, MA: MIT Press.

Kingsbury P & Palmer M (2002). 'From TreeBank to PropBank.' In *Proceedings of LREC–2002.* Las Palmas, Canary Islands, Spain. European Language Resources Association (ELRA), France.

Marcus M, Santorini B & Marcinkiewicz M A (1993). 'Building a large annotated corpus of English: the Penn Treebank.' *Computational Linguistics 19,* 313–330.

Meyers A, Reeves R, Macleod C, Szekely R, Zielinska V, Young B & Grishman R (2004). 'Annotating noun argument structure for NomBank.' In *Proceedings of LREC–2004.* Lisbon, Portugal. European Language Resources Association (ELRA), France.

Sampson G (1995). *English for the computer: The SUSANNE corpus and analytic scheme.* Oxford: Clarendon Press.

Sgall P, Hajicova E & Panevova J (1986). The meaning of the sentence in its semantic and pragmatic aspects. Dordrecht, Holland: D. Reidel Publ. Co.

Trier, Jost (1894–1970)

D Candel, CNRS, University of Paris 7, Paris, France

Jost Trier, the son of a medical doctor, was born in Schlitz. He studied German and Roman languages as well as comparative linguistics in Freiburg, Basel, Berlin, and Marburg. He received a professorship in German philology in 1932 at the *Westfälische Wilhelms-Universität*, Münster, where he became *Rektor* in 1956–1957 and *Prorektor* the following year. He became a member of the *Akademie der Wissenschaften* in Göttingen (1939) and was made *Ehrenmitglied* of the *Deutscher Germanistenverband* (1962) and of the *Institut für Deutsche Sprache* (1969). He was awarded the Konrad-Duden Prize (city of Mannheim, 1968), for having honored the German language with his work. Trier died in Bad Salzuflen in 1970, and his letters, manuscripts, and personal documents are stored at the University of Münster.

Trier, inspired by Saussure and influenced by Humboldt, developed, after Weisgerber's research, the lexical field method as applied to German etymology. His work *Der deutsche Wortschatz im Sinnbezirk des Verstandes. Die Geschichte eines sprachlichen Feldes* (1931), was the foundation for research in field semantics. He was an innovator in adopting a structuralist approach to studies of word meaning. He used each lexical item according to its semantic relations with the other items, to consider individual words and to group into lexical fields. As his interest was studying the whole system more than individual words, such an analysis was meant to cover the whole vocabulary: based on this, Trier chose small conceptual areas of vocabulary for his 'mosaic model'; for instance, the abstract notions of 'knowledge' in medieval German – *wîsheit*, *kunst,* and *list* – the former being a superordinate, the latter ones hyponyms, and, by the end of the 13th century, the structure of the field changed; *wîsheit* was no longer a generic, *list* had left the group, and *wizzen* had entered it.

The semantic field analysis shows that a conceptual region may change according to different languages or to successive states of a language. Trier's theory was criticized for considering vocabulary as a single