Treebank Development with Deductive and Abductive Explanation-based Learning: Exploratory Experiments

Oliver Streiter*
European Academy, Bolzano, Italy
ostreiter@eurac.edu

July 15, 2002

Abstract

In pace with the success of corpus-based approaches to theoretical and computational linguistics, the collocation of corpora has evolved into a research activity in its own. As the currently available corpora either lack annotation depth or closure, more data will be annotated in the future, preferably with minimal human intervention. This paper tries to approach the problem of treebank development from a logic-based learning perspective, applying several alternative forms of inference in order to assess their potential for automatically generalizing from a seed corpus annotated by hand to a corpus of POS annotated sentences, in order to automatically produce syntactic annotations that are good enough to use as training material for a parser. We shall show that syntactic annotations can be created automatically in large quantity via deductive and abductive explanation-based learning (EBL). Although these automatically created structures are not statistically representative with respect to many quantitative aspects of the treebank, the annotations may provide useful qualitative and quantitative data which might be extracted and reinvested into a parser. We shall compare the benefits and investments of automatically created structures to that of human-annotated structures and suggest some possible strategies how EBL approaches can be combined with manual annotation.

Keywords: treebank development, corpus-based parsing, explanation-based learning, deduction, abduction.

1 Introduction

With more and more computerized corpora becoming available, the development of techniques which allow to compile a corpus automatically into an NLP-tool has been established, together with the creation of such corpora, as a central issue in NLP. Monolingual corpora, for example, may be tagged for the part-of-speech (POS) in order to train POS taggers (e.g. 28; 6). These taggers may then in turn be used to develop larger tagged corpora. Corpora with richer syntactic annotations are commonly referred to as *treebanks*. They are developed for more and more languages, preferably from previously POS tagged corpora, (e.g. 3; 18; 13; 10; 15; 39; 14; 30).

*Most of the experimental work described here has been realized at CKIP (Chinese Knowledge Information Processing Group) at IIS (Institute of Information Science) at Academia Sinica in Taiwan. The author is currently affiliated to the European Academy where the drafting took place.

Due to the difficulties of natural language parsing, treebanks have been developed either completely manually, using a corpus-derived partial parser (21) or a hand-crafted parser, the output of which is manually post-editing. The post-edition may include the selection of the best parse (e.g. 3) and the correction of parsing errors (e.g. 5). In (33) we investigated how this work-intensive labor may be reduced. At this aims we staged a sequence of experiments which could show that when a parser can closely interact with the annotation tool (co-editing approach), annotations can be achieved significantly faster than when using the post-editing approach. We could equally show that in some specific contexts continuous feedback, i.e. training a parser with each freshly annotated sentence and phrases prior to parsing and correcting the next sentence or phrase, may reduce the manual annotation work due to improved parsing results.

In this paper we shall tackle the question in how far tree-structures may be acquired automatically from POS tagged corpora. Although there is the common agreement that the work of the human annotator will be irreplaceable for a long time, it is our conviction that much of the annotation work can be done with minimal human intervention in the near future. This paper tries to approach the problem of treebank development from a logic-based learning perspective, applying several alternative forms of inference in order to assess their potential for automatically generalizing from a seed treebank to a corpus of POS annotated phrases, in order to automatically produce syntactic annotations that are good enough to use as training material for a parser.

2 Experimental Design

2.1 Introduction

Given an existing (seed) treebank and a corpus-based parser which has been derived from it (c.f. 7; 4; 11; 34), one hypothetical way to increase the closure of the treebank is to parse sentences and store the parses in the treebank.[1] This is in fact how the paradigm of *explanation-based learning* (**EBL**) has been applied to Natural Language Processing. This paradigm aims at the transformation of a complex general knowledge resource (e.g. a grammar) into a specific and operational resource. Applied to the problem of parsing, sentences are parsed, the results stored and re-used in subsequent parses whenever possible (c.f. 25; 22; 31; 38). The results reported in these experiments attest a reduction of processing time and an ”acceptable” loss in the accuracy of the system.

In order to tackle the question whether a similar approach could also create new information which, for example, could result in an improved parsing accuracy, we shall set up a formal system which allows to reason about learning in the context of natural language parsing. We shall then come up with a set of hypothesis why learning beyond the improvement of run-time should be possible within the EBL paradigm (Section 2.4). The theoretical reflections will be tested in a row of experiments. In Section 3 we shall describe how tree-structures may be derived by deduction and abduction. In Section 4 we determine whether these tree structures contain new and representative information and thus whether we can speak of (successful) learning.

2.2 Formal Problem Description

In our formal notation, $[...]$ will be used to represent an ordered set. $\{...\}$ will be used to represent un unordered set. The $\#$ prefix to a set denotes the cardinality of a set. $!$ is used for the boolean operator ’not’ and $|$ as the boolean operator ’or’.

[1] For a discussion of the notion of the *closure* of a treebank c.f. (17).

We define the following entities. A theory $\mathcal{T}$ is a set $\mathcal{A}, \mathcal{L}, \mathcal{R}$ where $\mathcal{A}$ is a set of attributes a, $\mathcal{L}$ is a set of class labels l and $\mathcal{R}$ is a set of rules r. r, a and l may have an internal structure in the form of ordered or unordered sets of more elementary r, a and l respectively. Each r states how $\mathcal{A}$ and $\mathcal{L}$ may be related, e.g. $[a, l] = [[\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}], l]$. $\mathcal{O}$ is the set of observable data with each $o \in \mathcal{O}$ being a set $[a, i]$. i is a unique identifier of o_i. i may be decomposed as $i = [i^1, ..., i^n]$ or $i = \{i^1, ..., i^n\}$. $\mathcal{C}$ is the set of data classified according to $\mathcal{T}$, with $c = [o, l] = [a, i, l] \in \mathcal{C}$.

$\mathcal{T} := \mathcal{A}, \mathcal{L}, \mathcal{R}$	$r := [a, l] \| [o, l]$
$\mathcal{A} := \{a\}$	$o := [a, i]$
$\mathcal{L} := \{l\}$	$c := [o, l] = [a, i, l]$
$\mathcal{R} := \{r\}$	$r := [r_1, ..., r_n] \| \{r_1, ..., a_n\}$
$\mathcal{C} := \{c\}$	$a := [a_1, ..., a_n] \| \{a_1, ..., a_n\}$
$\mathcal{O} := \{o\}$	$l := [l_1, ..., l_n] \| \{l_1, ..., l_n\}$
$\mathcal{I} := \{i\}$	$i := [i_1, ..., i_n] \| \{i_1, ..., i_n\}$
$\mathcal{A} \in \mathcal{L}$	

If we transfer these definitions to the task of treebank development, we may conceive of $\mathcal{T}$ as a syntactic theory and $\mathcal{L}$ as the set of a syntax trees conform to $\mathcal{T}$. Phrase structure grammars and dependency grammars may be represented by ordered or unordered sets l. $\mathcal{A}$ is the tag set of words, phrases and sentences. a may thus correspond to a POS-tag. $\mathcal{O}$ is a corpus tagged with $\mathcal{A}$. $\mathcal{I}$ corresponds to a list of words, phrases or phrases (the surface string). $\mathcal{C}$ is the treebank where each tree c consists of features $a = [...]$, a sequence of lexemes $i = [...]$ and a syntax tree $l = [...]$.

According to the epistemological bias of $\mathcal{R}$, $\mathcal{R}$ may or may not be expressed in terms of $\mathcal{C}$. As each c by its definition includes a possible r, $\mathcal{R}$ may be partially or totally expressed by $\mathcal{C}$. This is the case for lazy learning approaches. They do not operate with rules of the type $r = [a, l]$ but $r = [[a, i], l]$. Eager learning approaches conceive of $\mathcal{R}$ as a competence grammar which is underspecified with respect to the data in $\mathcal{O}$ (c.f. 27). This underspecification is the result i) of the reduction of $[a_i, i]$ in c to $[a_i]$ (i is reduced to zero) or to $[a_{reduc}]$ with a_{reduc} != a_i and ii) the *universal generalization* from an existentially quantified statement to a universally quantified statement ($\exists(a_i, i)[a_i, i] \wedge [a_i, i, l]$ into $\forall(a_{reduc})[a_{reduc}] \rightarrow [a_{reduc}, l]$). For probabilistic grammars this is expressed as conditional probability ($P(l|[a_{reduc}]) = 0.45$). The reduction is motivated by frequencies observed in $\mathcal{C}$ which describe the contribution of i or parts of i as being of limited relevance for the assignment of l to o. Thus i or parts of i are reduced and rules are based on the same a_i or a new a_{reduc} which incorporates the unreduced part of i (a_{reduc} is a subset of a_i).

The process or reduction and universal generalization is referred to as *induction*.[2] Enriching $\mathcal{R}$ with relative frequencies of relations within $c = [[\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}], i, l] \in \mathcal{C}$ should allow, according to the eager learning approaches, to compensate for the loss of information caused by reductions when describing $\mathcal{O}$.

Continuing the transfer of the above model to natural language parsing, parsing may be conceived as a

[2]The notion of induction is due to the English philosopher Francis Bacon (1561-1626). Inductive logic is based on the assumption that a general law can be concluded by observing the relative frequency of events.

classification task in which $l \in \mathcal{L}$ is assigned to an instance $o \in \mathcal{O}$. Different from other classification tasks, natural language parsing requires an open set of target classes $\mathcal{L}$ (c.f 27). In order for such a classification to be possible, $\mathcal{L}$ and $\mathcal{R}$ have to be compositional and there must be a function from $\mathcal{R}$ to $\mathcal{L}$. This function is necessary so that for each application of r an corresponding l may be generated. For this reason, the $\mathcal{L}$ of treebanks is often expressed in terms of $\mathcal{R}$ (the label is a parse tree and not, for example, a number or a tag-label). Parsing thus differs from related approaches such as supertagging (c.f 32) in that for parsers $\forall(r \in \mathcal{R})r \rightarrow \exists(funct(r))funct(r) \wedge funct(r) = l$ while with supertagging, for example, only for an elementary subset of $\mathcal{R}$ there is such a function.

Parsing is a *deductive* inference if the assignment of l to o follows from r and $r = [a, l])$. This holds whether or not $\mathcal{R}$ is enriched with frequencies. Frequencies help to assign a unique l whenever $\mathcal{R}$ associates a disjunct l to a given $a \in o$. We shall see below that solving disjunctions in $\mathcal{R}$ remains with the deductions.

Parsing is based on *memory* if the assignment of l to o follows from r and $r = [a, i, l]$.

If the assignment of l to o follows from r_{abduct} and $!(r_{abduct} \in \mathcal{R})$ and $\exists(r)(r \in \mathcal{R})$ and there is a function from $a_{r_{abduct}}$ to $l_{r_{abduct}}$, parsing is an *abductive* inference.

Abduction subsumes reasoning by analogy. With analogy we distinguish a source domain of known entities ($c = r = [o, l]$) and a target domain ($[o', l']$) with l' being unknown. The object to be classified o' is classified as $l' = l$ or as $l' = funct(l, o, o')$. In order for this abduction to qualify as analogy the same predicates $p_1 \ldots p_n$ have to apply to o or its components and o' or its components (thus describing a similarity). Thus if the assignment of l to o follows from r_{abduct} and $!(r_{abduct} \in \mathcal{R})$ and $\exists(r)(r \in \mathcal{R})$ and there is a (similarity)-function $a_{r_{abduct}}$ to $l_{r_{abduct}}$, parsing is achieved via analogy.

2.3 Abduction

The notion of *abduction* has been introduced by the Charles S. Peirce (1839-1914). Abduction is a process of hypothesis generation. The generated hypothesis may help to achieve a preliminary classification. Deduction and abduction may be used conjointly whenever deductive inferences encounter gaps. A deductive inference stops in front of a gap and does not come to a conclusion. An abduction creates a new hypothesis which allows to bridge the gap and to continue the inference. The correctness of the abductive inference, unlike the deductive inference, does not depend on the correctness of the generated hypothesis, but on the correctness of the instantiated generated hypothesis: Imagine, you ordered a product and, one week later, you receive a parcel. Using abduction you might assume that this parcel contains what you ordered. In order to come the this conclusion you induce from single experiences $\exists(x, y)order(x, y) \wedge receive(x, y)$ a hypothesis $\forall(x, y)order(x, y) \rightarrow receive(x, y)$ and instantiate x and y with "I" and "product", so that you (safely) assume that you receive in the parcel the ordered product.

There are different kind of gaps and different techniques to bridge them. Most of them can be reduced to either the universal generalization or the reduction. Abduction is deliberately based on a limited set of observables, so as to quickly come to a conclusion. As a hypothesis, we might thus classify all local lazy learning approaches, which try to interpolate from or combine stored training data close to the hypothetical solution as abductive devices. The functions from $a_{r_{abduct}}$ to $l_{r_{abduct}}$ are adaptation functions which either assign the same l to a and $a_{r_{abduct}}$ (with universal generalizations or reductions), or such an l' to ar_{abduct} that the difference of l' to l can be derived from the difference between a and ar_{abduct} (with reductions).

Variable instantiation: A first kind of abduction is the *variable instantiation*. Imagine the following syllogism (P_1 to P_n are the premises and C_n is the conclusion).

1: P_1: Michael wanted to go to Manila.

P_2: Someone is calling me from Manila.

C_1: Michael is now in Manila.

Here C_1 is reached by instantiation the existentially quantified variables (TIME$_{P_1}$) and (PERSON$_{P_2}$) with the instances of the other preposition (TIME$_{P_1}$=now$_{P_2}$) and (PERSON$_{P_2}$=Michael$_{P_1}$) (c.f. 23). If a variable is universally quantified, the instantiation remains with the deductions, if the variable is differently quantified, the instantiation belongs to the abductions. This abduction does not qualify as analogy.

Term identification: A similar unification (identification) may operate on terms, assuming that different terms possibly stand for the same entity (either the entity has more than one name, or one name is misspelled, or translated, or incorrectly remembered etc). Two terms may be equally identified if we assume them to have similar consequences, occur in similar contexts etc. (19), for example, suggest to identify *jogging* and *horse racing* when trying to explain the sudden death of a sportsman. In other words, rules are allowed to apply when $a_i \in o \neq a_j \in r$ and there is a relation $rel()$ such that $rel(a_i, x_i)$ and $rel(a_j, x_j)$ and $x_i = x_j$. We see that term identification can be reduced to a variable unification in which an existentially quantified variable (someone, something) (e.g. refer("Michel",someone) or share_consequences("horse_racing",something)) is instantiated with "Michael" or "jogging".

2: P_1: Michael wanted to go to Manila.

P_3: Someone called "Michel" is calling me from Manila.

C_2: Michael is now in Manila.

This reasoning step can be easily reduced to an abduction. P_1 is a piece of background knowledge which motivates the the term identification. "Michel" and "Michael" are o and o' respectively which are related via similarity functions. As a consequence l' is set to l:

2': P_3': "Michel" is now in Manila.

C_2: Michael is now in Manila.

Deletion: When the entire set of $a_r = [a^1, ..., a^n] \in r$ does match no $a_o \in o$, we may delete some $a^m \in a_r$ so that $a_r = a_o$ (the rule may apply). Deleting $a^m \in a_r$ with a_r being an unordered set $\{a^1, ..., a^n\}$ equally belongs to this abduction technique (c.f. C_3). This operation can be explained neither as reduction nor as generalization and is maybe the most risky type of abduction.

3: P_4: If X can sing and fly it is a bird.

P_5: Rockey can fly.

C_3: Rockey is a bird.

Reduction Omitting some $a^m \in a_o$ with $a_o \in \mathcal{O}$ so that $a'_o = [a^1, ..., a^n] = a_r$ represents an additional abduction technique. If $a_o \in O$ is not ordered however, omitting a^m from $a_o = \{a^1, ..., a^n\}$, remains with deductions (c.f C_5).

4: P_6: If X buys a house, X has a house.
P_7: Rockey buys and sells a house.
C_4: Rockey has a house.

5: P_8: If X can fly it is a bird.
P_9: Rockey can fly and sing.
C_5: Rockey is a bird.

2.4 Learning

A system is learning when its performance improves over time. In order to make learning possible, internal states of the system which influence the performance have to change. Thus, although nothing can be learned from a deductive inference from logical point of view (logic is concerned with truth values and not with knowledge), new states may result from the *memorization* of $c = [a, i, l]$ in $\mathcal{C}$ and the usage of $\mathcal{C}$ within $\mathcal{R}$. New r_{new} may be derived from new c by induction which transforms $\exists(i)[a, i] \wedge [i, l]$ into $\forall(a)a \rightarrow l)$. For successful learning to take place $\exists(i)[a, i] \wedge [i, l]$ has to contain new information. Learning may also take place if the probabilities associated with r are refined. For successful learning to take place, the new probabilities should be more representative for the data in $\mathcal{O}$ than the previous probabilities had been.

Format: With deductions, rules r_{new} may be new in the sense that already existing rules have a different format. In the case of EBL, for example, a more operational representation format is aimed at. No matter how complex the original rule-format is, rules derived from $\mathcal{C}$ have the format $[[\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}], l]$. If, for example, a complex HPSG-grammar is partially transformed through EBL into tree-structure with instantiated values (c.f. 22), the matching of $[\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}]$ onto the input structure replaces typed-feature unification. Thus learning consists of the acquisition of rules of the type $r = [a, l]$ where previously more complex rules have been used. With this kind of learning, only improvement in the processing time has been achieved.[3]

Macros: The rules may be also new in the sense that such a rule is literally absent in $\mathcal{R}$ although implied by the combination of more than one $r \in \mathcal{R}$. This is the case of *macro learning* (e.g. 37). A macro r_{macro} is a rule, the application of which yields a result which can be obtained equally via the application of a set of rules $\mathcal{R}'$ with $\#\mathcal{R}' \geq 2$ and $!(r_{macro} \in \mathcal{R}')$. In terms of a grammar, such macros correspond to redundant phrases, i.e. phrases which are obtained by composing smaller phrases which form equally part of $\mathcal{R}$. Redundant phrases are necessarily compositional phrases. Compositional phrases however are only redundant phrases if their components are elements of $\mathcal{R}$. Macros represent a shortcut in the maze of phrase combinations and learning takes place through the reduction of ambiguity. Macros may also effect the probability assigned to a parse tree as the probability of the macro may be different from the product of the probabilities of its components (e.g.

[3]E.g., translating a Russian novel into Chinese, allows the Chinese reader to read the novel faster. A certain loss of information however cannot be avoided and no new information is added.

if the components are infrequent but the macro relatively frequent, or if the components are frequent but the macro relatively infrequent.)

Statistics: Specific relations of a_i, i_i and l_j in $\mathcal{C}$, may be denoted as $rel(a_i, i_i, l_j)$. When establishing the cardinality $\#rel(a_i, i_i, l_j)$ or the probability $prob(rel(a_i, i_i, l_j))$ we obtain new system parameters which may allow to improve the performance (e.g. 7; 27).

Attractiveness: With deductions, in addition, until then uncovered relations $[i, l_i]$ may be acquired from $c = [a_i, i_i, l_i]$. The acquired sets $c = [a_i, i_i, l_i]$ may also be a valuable resource in a memory-based model as they may increase the attractiveness of the class represented by l_i. The attractiveness of l_i for a or i is a function of the cardinalities of a_{l_i} or i_{l_i} in l_i respectively, i.e. $attract(l_i, a_{l_i}) = funct(\#l_i, \#a_{l_i})$ and $attract(l_i, i_{l_i}) = funct(\#l_i, \#i_{l_i})$. We may think of the attractiveness in absolute terms, e.g. as a token type relation $\frac{\#i_{l_i}}{\#l_i}$ or $\frac{\#a_{l_i}}{\#l_i}$, or in relative terms, as attraction with respect to a certain o to be classified, e.g. via the IF.IDF-function or the cosine similarity function between o and the centroid of l_i (c.f. 24).

It is thus easy to see how the attractiveness changes with new c. With abductions, $\#[a_i, l_i]$ changes from 0 to $x \geq 1$. With deductions, $\#[i, l_i]$ increases from $x \geq 0$. After learning an abduction, new deductions are possible which are attracted by the new a_i. After learning a deduction, different results may possibly obtained due to a modified attractiveness $attract(l_i, i)$.

In order to illustrate the concept of attractiveness, imagine two classes l_j ('bird') and l_k ('fish') each to be represented by one instance $o_j \in \mathcal{O}$ ('sparrow') and $o_k \in \mathcal{O}$ ('tuna fish'). Classification errors may easily occur. If we add more o to either l_j or l_k, (e.g. 'ostrich' and 'swan' to the 'birds', and 'herring' and 'shark' to the 'fishes'), classification errors, e.g. for 'penguins' and 'flying fishes', become less likely due to the improved attractiveness with respect to properties like 'big' and 'swim' for birds and 'small' and 'fly' to the fishes (c.f. 26).

All types of learning which are possible with a deduction are also possible for a correct abduction. In addition, new relations $[a_i, l_i]$ and thus new r_{abduct} may be learned from $c = [a_i, i_i, l_i]$. While with deductions r_{new} is new in the sense that it has a new format, represents a macro or an improved representativeness, an abduced r_{abduct} represent new relations $[a, l]$, which are not covered by $\mathcal{R}$. Abduction thus enlarges $\mathcal{R}$ by those r_{abduct} which are not in $\mathcal{R}$, which cannot be deduced from $\mathcal{R}$ and which are necessary in order to describe $\mathcal{O}$. We shall refer to such r_{abduct} as *relevant* for learning.

2.5 Parsing

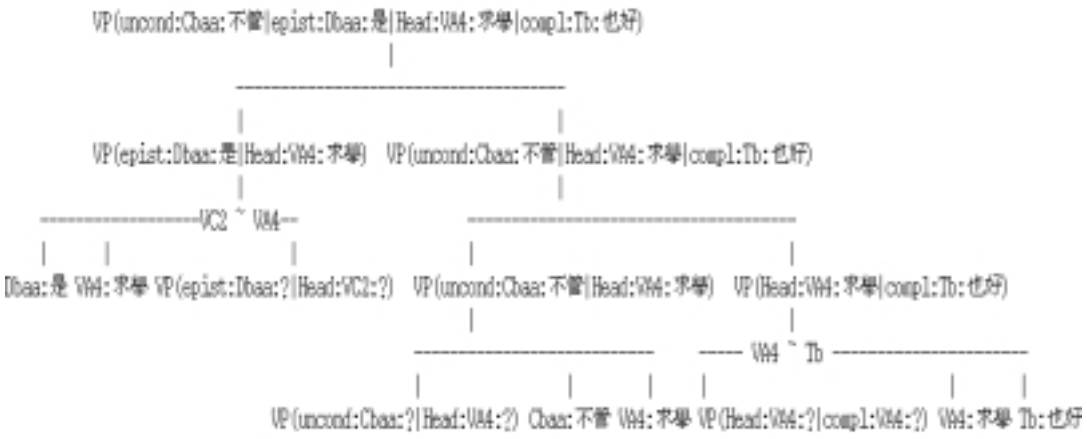
Having seen how learning from deductions and abductions is theoretically possible, we automatically create new tree structures by parsing a Chinese corpus $\mathcal{O}$ with the help of an example-based Chinese parser. This corpus of 493.000 sentences and phrases represents that part of a 5.000.000 word Chinese Corpus (c.f. 16) for which the extended POS labels used in the treebank could be automatically reconstructed from the reduced tag-set used in the 5.000.000 corpus. Due to this pre-selection, the 493.000 structure corpus shows a reduced structure length (5.9 words per structure vs. 6.27 words per structure).[4]

The example-based parser OCTOPUS used for this parsing task (35) integrates inferences based on memory, deductions and abductions (36). This parser is trained on a 10.000 tree structures ($\mathcal{C}_{seed}$)

[4]The relative short structure length is due to the applied standards of word segmentation and sentence segmentation and the wide range of text types, including originally spoken dialogues (9).

before parsing $\mathcal{O}$. For every $o \in \mathcal{O}$ the parser produces one parse-tree l and one *explanation*. The explanation has the form of a derivation trees as found in logical proofs or with Tree Adjoining Grammars (c.f 29). The explanation forms either a connected tree or disconnected branches of the derivation tree which are linked via abductions. For example, OCTOPUS may unify terms if surface similarities and contextual constraints support this identification (c.f. Fig. 1). As deduction and abduction steps are visible in the explanation, we know by which inference steps the tree structure has be generated.

Figure 1: An explanation produced by OCTOPUS. Above we show the final parse which is recursively deduced from two or more more elementary premises. Gaps in the deduction are bridged in this example with the abduction technique of *term identification* (X $\sim$ Y). The marker '?' is a graphical shortcut for the set of lexemes $\{i\}$ which have occurred in l in the same position.



3 Experiments

Parsing $\mathcal{O}$ yields the corpus $\mathcal{C}_o$. With the help of the explanation generated for each $c \in \mathcal{C}_o$ we can filter out different sub-corpora which represent different inference types. Analyzing the properties of these sub-corpora, we may judge the usefulness for each kind of inference for the automatic learning of tree-structures.

3.1 Deducing Trees without Recursion

The first filter extracts from $\mathcal{C}_o$ those c which are deduced in one single non-recursive deduction step. This results in a corpus $\mathcal{C}_{D_1}$ with 29.300 tree-structures.

> If there is a $r_i \in \mathcal{R}$ with $r_i = [a_i, l_i]$ and an $o \in \mathcal{O}$ with $o_j = [a_j, i_j]$ and $a_i = a_j$, then o_j is assigned the label l_i in $\mathcal{C}$ so that $c_j = [a_j, i_j, l_i]$.

3.2 Deducing Trees with Recursion

Using a second filter we extract an additional corpus of 700 tree-structures which conform to a recursive deduction.

If there is $r_j = [a_j, l_j] \in \mathcal{R}$ and there is $o_k = [a_k, i_k] \in \mathcal{O}$ such that a_j is contained in $a_k = [..., a_j, ...]$ we say a_j matches a_k. If the matched part in a_k is replaced by a_{l_j}, which is the representative of l_j, thus creating a new $a'_k = [..., a_{l_j}, ...]$ and if there is $r_k = [a'_k, l'_k]$, l_k is obtained by replacing the representative a_{l_j} in l'_k by l_j.

3.3 Abduction by Term Identification

A third filter retains those structures which were obtained by maximally one identification of POS-labels. In order to minimize possible parsing errors, the filter requires that only the last characters of the term may change (we thus accept the identification of the `Nab` and `Nac`, but not of `VC2` and `VA4` as shown in Fig. 1). The corpus $\mathcal{C}_{A_1}$ contains 7.273 new tree-structures.

A_1: If there is a $r_i \in \mathcal{R}$ with $r_i = [a_i, l_i]$ and an $o \in \mathcal{O}$ with $o_j = [a_j, i_j]$ and all but one $a_{l_i}^{k_i}$ in $a_i = [\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}]$ are identical to $a_{l_j}^{k_j}$ in $a_j = [\{a_1^1, ..., a_1^n\}, ..., \{a_n^1, ..., a_n^n\}]$ and the $a_{l_i}^{k_i}$ and $a_{l_j}^{k_j}$ differ with respect to the rightmost character and $i_i \neq i_j$, then o_j is assigned the label l_i so that $c_j = [a_j, i_j, l_i] = [o_j, l_i]$.

4 Properties of Deduced and Abduced Trees

The evaluation of the automatically deduced and abduced tree structures cannot resort to the correctness of these trees, as we lack any standard in order to judge the correctness of these trees. In addition, the correctness of the trees is not the only property we have to consider. Equally important is the question whether an automatic applications X which operates on $\mathcal{C}_{seed}$ can be improved when $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$ and $\mathcal{C}_{A_1}$ are added to the training data. We therefor try to find out whether deduced and abduced trees contain new and representative information.

We compare the properties of the derived corpora ($\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$, $\mathcal{C}_{A_1}$) to that of $\mathcal{C}_{seed}$, $\mathcal{C}_{hand}$ and $\mathcal{C}_{ref}$. $\mathcal{C}_{seed}$ is a randomly selected subset of the entire treebank used for training purpose. The properties of $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$, $\mathcal{C}_{A_1}$ and $\mathcal{C}_{hand}$ thus should be similar to those of $\mathcal{C}_{seed}$ if these corpora are to be considered good corpora.

$\mathcal{C}_{hand}$ is a corpus of 1000 human-annotated structures, a three month production at CKIP. Basically, $\mathcal{C}_{hand}$ has two properties. It is corrected by humans and thus is as correct and consistent as human work can be. Secondly, this corpus is not representative for the treebank, as it is obtained by processing one article after another, while the treebank is a balanced collection of various articles. Although we cannot distinguish these two properties of $\mathcal{C}_{hand}$ we can compare automatically learned corpora ($\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$, $\mathcal{C}_{A_1}$) and $\mathcal{C}_{hand}$ to the properties of $\mathcal{C}_{seed}$ and, secondly, we can compare $\mathcal{C}_{hand}$ to the automatically learned corpora.

$\mathcal{C}_{ref}$ finally are 10.000 randomly selected untrained tree structures from the treebank.

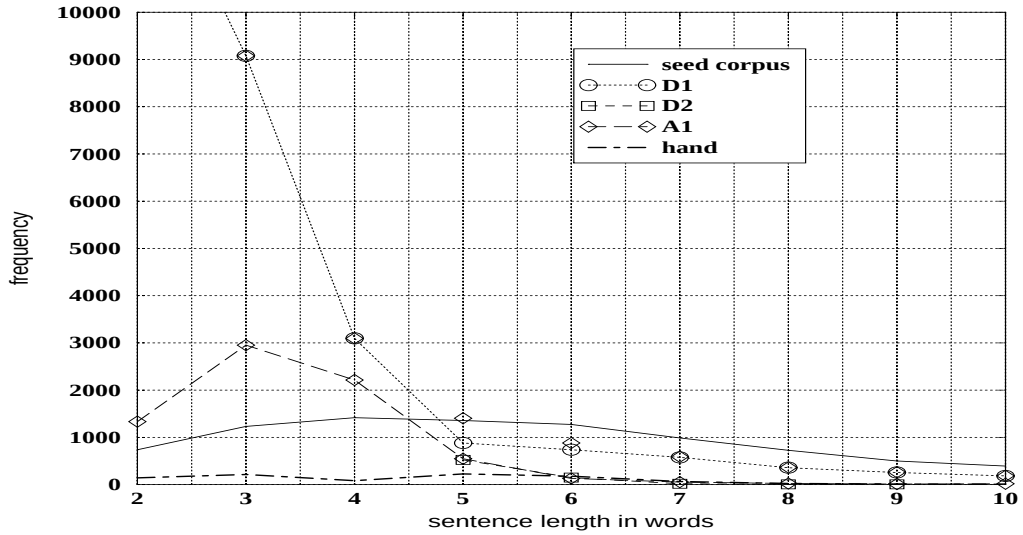
How is this comparison of different corpora to take place? As possible means of comparison we shall look i) at the structure length in these corpora, ii) at the frequencies of main phrasal nodes, iii) at the frequencies of word-to-word relations, iv) at the learned macros, v) at the usefulness as training material for the corpus-based parser OCTOPUS and vi) at the changes in attractiveness. (i-ii) are rough and weak indicators which allow to judge the representativity of the automatically trained corpora. If these indicators do no show a difference, nothing follows from it. However if they show a difference, then the automatically trained corpora are not representative. (iii) is a measure which allows to judge the representativity and the presence or absence of new information. (iv) allows to determine whether

new information could be acquired .(v) uses an example-based parser as one possible application the quality of the output of which depends on the quality of the input treebank. (vi) looks again at the acquisition of new information, this time from the perspective of the attractiveness of l.

4.1 Structure Length

As the deduced tree-structures of $\mathcal{C}_{D_1}$ are obtained by identifying $a_{l_i}^{k_i}$ in $\mathcal{O}$ and $\mathcal{C}$ and identity is more likely with short a, most tree-structures in this corpus are short. The distribution of structures of different lengths is thus extremely distorted (c.f Fig. 2). The deduction with a recursive theory does not improve the representativity of the deduced data with respect to the structure length. The abduced trees in $\mathcal{C}_{A_1}$ show a similar distortion as those in $\mathcal{C}_{D_1}$, although less pronounced. The reason for the distortion in $\mathcal{C}_{A_1}$ is similar to that of $\mathcal{C}_{D_1}$: $\mathcal{C}_{D_1}$ is produced by an overlap in $\mathcal{O}$ and $\mathcal{C}$, although the conditions for the overlap have been relaxed. It thus seem that if we want to control the representativity of deduced and abduced trees, we have to cope with the bias of short structures. On the other hand we see that the number of structures which can be created automatically in a few hours is larger than the number of structures which can be obtained in several years of manual annotation.

Figure 2: The frequency of structures of different length for $\mathcal{C}_{seed}$, $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$ and $\mathcal{C}_{A_1}$.



4.2 Phrasal Nodes

The proportions with which the main phrasal nodes occur helps to typify the content of the corpora. In Tab. 1 we reproduce the proportion of the main phrasal categories NP, PP and VP (the latter including both VP and S). In addition, we reproduce the proportion with which these phrasal categories are the highest node in the structure (marked as Top-NP etc), thus typifying the whole structure as NP, PP etc.

Tab. 1 shows that all deduced and abduced corpora contain an un-proportionally large part of Top-NPs, followed by an un-proportionally large part of Top-PPs. When looking at the total distributions of phrasal categories, the abduced corpus reflects the frequency relations of $\mathcal{C}_{seed}$ most reliably. When comparing automatically learned corpora and $\mathcal{C}_{hand}$ we see that $\mathcal{C}_{hand}$ is much more representative

Table 1: The percentage of phrasal categories in deduced, abduced and hand-coded corpora in comparison to the percentage found in the seed corpus $\mathcal{C}_{seed}$.

	$\mathcal{C}_{D_1}$	$\mathcal{C}_{D_2}$	$\mathcal{C}_{A_1}$	$\mathcal{C}_{hand}$	$\mathcal{C}_{seed}$
PP	0.06	0.03	0.07	0.07	0.08
NP	0.30	0.34	0.35	0.27	0.34
VP	0.49	0.31	0.40	0.48	0.42
Top-PP	0.05	0.00	0.08	0.035	0.04
Top-NP	0.23	0.51	0.26	0.16	0.13
Top-VP	0.68	0.48	0.57	0.76	0.82

with respect to the Top-categories. However, as for the overall distribution, the automatically learned corpora are equally or more representative.

There are two possible reasons why NPs are learned more easily than verbal phrases and sentences. First, NPs tend to be shorter than VPs and Ss. Secondly, NPs show less variance in their internal structure so that they are matched more easily.

4.3 Word-to-word Relations

Word-to-word relations (two ordered words, disambiguated by their part of speech, related by a directed labeled graph, e.g. *tā*:pro $\overleftarrow{agent}$ *kàn:v7*, or *kàn:v7* $\overrightarrow{theme}$ *tā:pro*) are considered to be powerful information units which can be drawn from treebanks. Most prominent models which build on such units are as so-called *Immediate-Head Parsing Models* (e.g 12; 8; 11; 2). As data on word-to-word relations are sparse, deductive and abductive learning might offer a way to improve such models.

Tab. 2 reproduces the number of word-to-word relations extracted from the corpora. In order to test the representativity of these new relations, we compared the prediction of the frequency of word-to-word relation for $\mathcal{C}_{ref}$. At this aims we calculate the correlation of the predictor (frequency of a word-to-word relation in $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$, $\mathcal{C}_{A_1}$, $\mathcal{C}_{hand}$ and $\mathcal{C}_{seed}$) and the data to be predicted (the frequency in $\mathcal{C}_{ref}$). In a first comparison we evaluate the prediction made by the entire corpora. As the size of the predicting corpora obviously helps to make correct predictions, we also calculate the prediction made by 1000 relations of each of the learning corpora, taking the mean of 5 random samples. In a third comparison we calculate the correlation, using the Good-Turing estimator in oder to estimate the frequencies for unseen events and to smooth the seen events accordingly (c.f. 20).

Table 2: The number of word-to-word relations (above) and the correlation of the frequencies of word-to-word relations with the frequency of word-to-word relations in an unseen corpus.

	$\mathcal{C}_{D_1}$	$\mathcal{C}_{D_2}$	$\mathcal{C}_{A_1}$	$\mathcal{C}_{hand}$	$\mathcal{C}_{seed}$
total number	39.200	1500	10.200	1.874	25.300
correlation total	0.54	0.23	0.39	0.35	0.6291
correlation 1000	0.037	0.117	0.093	0.160	0.164
correlation Good-Turing	0.3679	0.2	0.3455	0.2936	0.6344

Tab. 2 reveals that the frequencies of word-to-word relations in deduced and abduced corpora are biased in the sense that certain word-to-word relations occur relatively frequently or infrequently. The data thus are not representative.

When looking at the predictions made by 1000 relations, we see that the representativity of the automatically learned corpora falls behind that made by hand-coded corpora. Due to the high number of word-to-word relations which can be obtained via automatic learning however, astonishingly good predictions can be achieved when using the entire corpus. We see that the worst predictor $\mathcal{C}_{D_1}$ who has the highest number of relations comes closest to the ideal predictor $\mathcal{C}_{seed}$.

Only the correlation of the frequencies of $\mathcal{C}_{seed}$ improve when the Good-Turing estimation is applied. This, once again shows that only these data are balanced and representative.

It thus may be possible to profit from the high number of biased word-to-word frequencies in the automatically learned corpora in order to improve the prediction made by the available corpora ($\mathcal{C}_{seed}$). In order to show this, we compare the prediction made by $\mathcal{C}_{seed}$ to that made by $\mathcal{C}_{seed}$ plus the deduced and abduced corpora. The prediction of $\mathcal{C}_{seed}$ plus the additional corpora is the (smoothed) interpolation of the frequencies of the two corpora. In the first comparison we give the same weight (0.5) to $\mathcal{C}_{seed}$ and the additional corpora. In a second comparison we assign the weight $1 - \frac{1}{f+1}$ to the frequencies of $\mathcal{C}_{seed}$ and $\frac{1}{f+1}$ to the frequencies of the additional corpora, where f refers to the frequency which is to be interpolated.

The results are reproduced in Tab. 3 show that the predictions of $\mathcal{C}_{seed}$ improve when combined with automatically learned corpora. The data further show that automatically learned corpora require a treatment (filters or weights) which are different from hand-coded trees. When comparing finally the improvement with automatically learned corpora with that obtained from human-annotated structures we easily see that abduced and deduced corpora fall behind the improvement obtained from hand-coded trees.

Table 3: The correlation of the frequencies of word-to-word relations with the frequency of word-to-word relations in an unseen corpus when seed-corpus and additional corpora are combined via an interpolation. λ_s refers to the weight assigned to the seed-corpus and λ_a to the weight assigned to the additional corpora.

	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$	$\mathcal{C}_{seed}$
$\lambda_s = \lambda_a = 0.5$	0.6210	0.6383	0.6367	0.6618	0.6344
$\lambda_s = 1 - \frac{1}{f+1}$ and $\lambda_a = \frac{1}{f+1}$.	0.6394	0.6399	0.6403	0.6439	0.6344

4.4 Macro Learning

It is relatively easy to establish whether or not macro learning takes place: First of all we may distinguish lexical macros ($r = [o, l]$) from categorial macros ($r = [a, l]$). While the first require a match at lexical level (i), the latter apply also when only POS-tag (a) match. Secondly we can distinguish potential macros from applied macros. Potential macros are those macros in the corpus which conform to our definition of macros. Applied macros are those potential macros which are matched in an unseen reference corpus. The difference between these is easy to see with $\mathcal{C}_{D_2}$. All tree structures of this corpus are by definition (potential) macros. However, none of these macros is applied in practice, due to the length of at minimum 5 words. $\mathcal{C}_{D_1}$, on the other hand can by definition contain no categorial

macros, while lexical macros may well be acquired (although their practical value is very limited).

Table 4: The number of categorial and lexical macros and their average length (in number of words per macro).

potential cat. mac.	$\mathcal{C}_{seed}$	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$
number of macros	14086	14086 (0)	15087 (+930)	15634 (+1548)	15662 (+1576)
macro length	4.94	4.83 (-0.11)	4.94 (0)	4.77 (-0.17)	4.96 (+0.02)
potential lex. mac.	$\mathcal{C}_{seed}$	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$
number of macros	15005	15147 (+142)	16006 (+1001)	17824 (+2819)	16681 (+1676)
macro length	4.83	4.81 (-0.02)	4.73 (-0.1)	4.98 (0)	4.56 (-0.27)
applied cat. mac.	$\mathcal{C}_{seed}$	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$
number of macros	1143	1143 (0)	1143 (0)	1416 (+173)	1210 (+67)
macro length	3.37	3.37 (0)	3.37 (0)	3.33 (-0.04)	3.38 (+0.01)
applied lex. mac.	$\mathcal{C}_{seed}$	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$
number of macros	206	206 (0)	207 (+1)	213 (+7)	231 (+25)
macro length	3.73	3.73 (0)	3.73 (0)	3.71 (-0.02)	3.74 (+0.01)

Table 4 shows that macro learning is practically absent with deductive EBL. When comparing abductive EBL with hand coding, we see that although hand-coded trees contain more macros, those acquired with abductive EBL have a higher recall, e.g. they are more likely to be applied. Hand-coded trees seem to provide longer macros than automatically learned trees. This may be of importance, as the longer the macros the more accurate the matches (c.f. 34).

4.5 Training the Parser

In order to investigate whether the new tree-structures (the new macros) may be used to support further parsing and the treebank development, we train the parser OCTOPUS, trained with $\mathcal{C}_{seed}$, with the additional training corpora $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$, $\mathcal{C}_{A_1}$ and $\mathcal{C}_{hand}$. The parsing accuracy is tested on 1000 untrained, unrelated, randomly selected structures (i.e. testing the coverage as defined in (35)) using as measure the f-score on dependency relations (c.f. 34). Fig. 3 reproduces the mean values obtained from a 5 times repeated random sampling.

OCTOPUS is a lazy learner. Lazy learners do not create an a priori intensional concept description as eager learners do but classify entities with the help of a subset of the stored training data (1). These *local* approaches, which try to interpolate from or combine stored training data close to the hypothetical solution, do not necessarily benefit from probabilities calculated over the whole training set. OCTOPUS, for example, uses the attractiveness of example-trees to the input structure in order to select appropriate example-trees. When training OCTOPUS with deduced and abduced trees, changes in the attractiveness of l take place. As the number of a_i or i_i in l_i influence the attractiveness, probabilities are learned implicitly in the form of a relative attractiveness of l. In principle however OCTOPUS is insensitive to the frequency of a structure when retrieving the most similar example structure.

The acquisition of macros might also improve the performance of OCTOPUS as the probability of incorrect phrase composition is reduced if macros are acquired from short tree structures with low

ambiguity (e.g. NPs) and used in a long structure with high ambiguity (for a save NP recognition).

Figure 3: The parsing accuracy (f-score of correctly identified dependency relations) with 1000 additional training trees from a human-annotated corpus, $\mathcal{C}_{D_1}$, $\mathcal{C}_{D_2}$ and $\mathcal{C}_{A_1}$. Left the exact curve, right the linear regression.

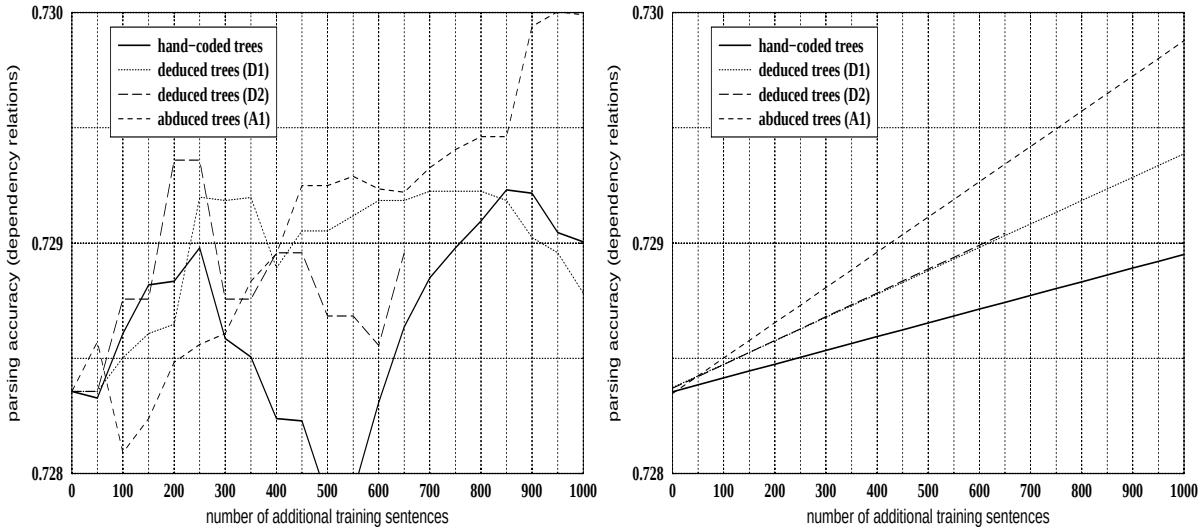


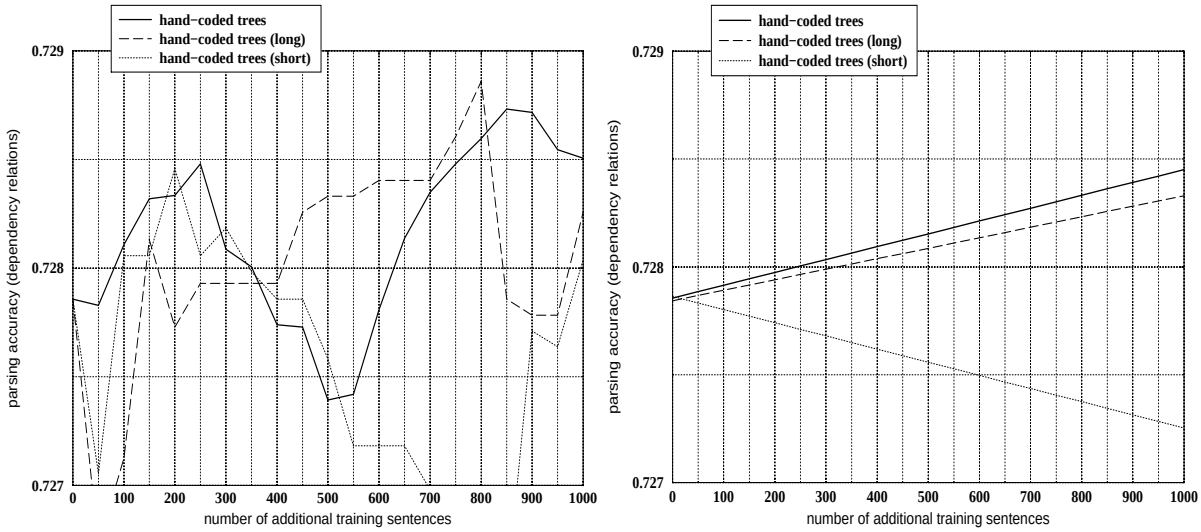
Fig. 3 shows that it is possible for a parser to learn from deduced tree-structures. This is a fact that previous research on EBL in NLP has neglected. Surprisingly, there seems to be no great difference in the gain from deduced and human-annotated corpora. Abduced corpora seem to be a better training material for the parser than either deduced trees or human-annotated trees. These data conform to our observations with learned macros. However we cannot explain the improved performance with deduced structures as here macro learning is almost absent.

While the improvement with abduced corpora is easy to understand as with new relations $[a, l]$ relevant information is acquired, the success of deduced trees is more difficult to understand. After all, all ways of learning with deduced corpora are also possible with human-annotated corpora and the latter are less biased.

In order to understand what influences the parsing results, we split the human-annotated additional training corpus into two corpora of the same size, one containing the shorter structures, one containing the longer structures and using these corpora as additional training material (Fig. 4). As the results show, training only short structures causes the parsing results to decrease. The improvement of the parser is almost exclusively due to the properties of the relatively longer structures. Again, this supports the importance of macro learning, but cannot explain the gain in parsing accuracy with deductions.

Fig. 5 shows the gain in parsing accuracy resulting from $\mathcal{C}_{A_1}$ as additional training material. This is compared to the human annotation efforts which is required in order to obtain the same results (represented by structures number 10.000 to 20.0000 of the treebank). With 4000 structures being a average estimate of the collective human year production of trees at CKIP, we see that a huge human effort would be required in order to obtain similar increases in the parsing accuracy once 10.000 trees have been encoded. This result implies that the future encoding strategy should shift its focus from human-driven encoding to machine-driven encoding with human support.

Figure 4: The parsing accuracy with short and long structures as training material. Left the exact curve, right the linear regression.



4.6 Attractiveness

We approximate the attractiveness of $\mathcal{L}$ by dividing total of all types of o by all types of l. We calculate this value for $\mathcal{C}_{seed}$ and for all $l \in \mathcal{C}_{seed}$ for $\mathcal{C}_{seed+D_1}$, $\mathcal{C}_{seed+D_2}$, $\mathcal{C}_{seed+A_1}$ and $\mathcal{C}_{seed+hand}$.

Table 5: The attractiveness of $\mathcal{L} \in \mathcal{C}_{seed}$ for the union of $\mathcal{C}_{seed}$ and the additional corpora. Theses data are split according to the length $length()$ (counted in words) of the structures.

	$\mathcal{C}_{seed+D_1}$	$\mathcal{C}_{seed+D_2}$	$\mathcal{C}_{seed+A_1}$	$\mathcal{C}_{seed+hand}$	$\mathcal{C}_{seed}$
all structures	2.64	3.30	2.19	2.26	2.15
$length() < 4$	4.22	5.52	3.34	3.47	3.26
$3 < length() < 7$	1.09	1.11	1.07	1.07	1.06
$6 < length()$	1.00	1.00	1.00	1.00	1.00

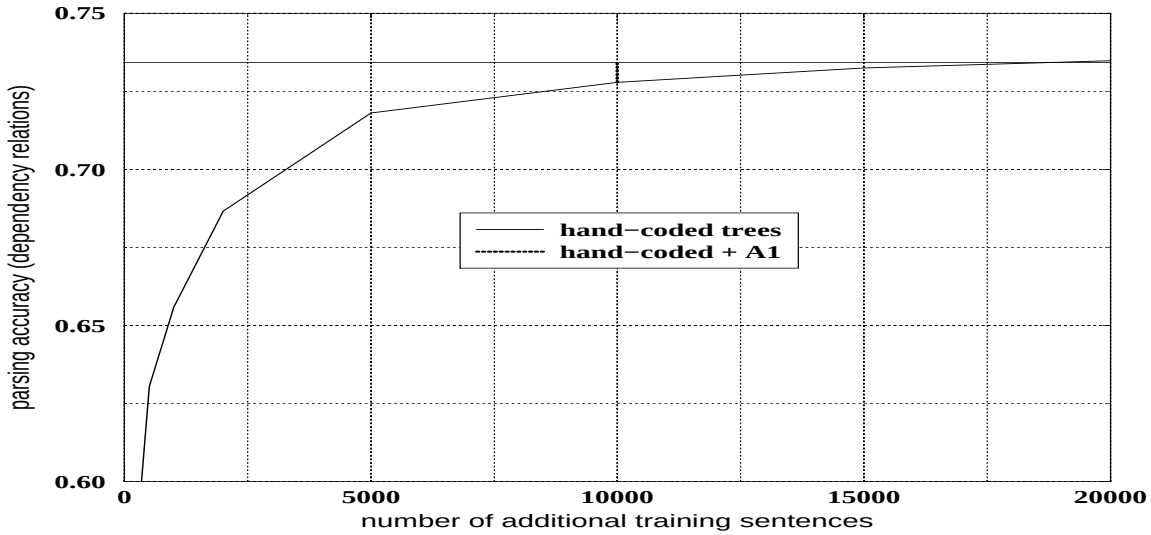
Fig. 5 shows that learning in the form of an increasing attractiveness of different l takes place. However, this gain is greatest with short structures. The attractiveness thus cannot be used in order to explain the improvement with deduced trees observed in the parsing experiment.

5 Summary

In this paper we have focused on the automatic acquisition of syntactically annotated corpora given a seed treebank and an annotated corpus. We introduced a terminology which allows to describe deductive and abductive learning for classification tasks such as natural language parsing. Using this terminology we hypothesized that it should be possible to learn from deductions and abductions.

Using controlled steps of deduction and abduction we produced new corpora. Our main concern was to find out whether the information contained in the deduced and abduced trees is new and representative

Figure 5: The gain in parsing accuracy from adding $\mathcal{C}_{A_1}$ to $\mathcal{C}_{seed}$ (10.000 structures) compared to continuing with human annotation.



and how the new information can be used.

We found that deduced tree-structures are biased with respect to most of their statistical properties. This fact does not depend on whether or not a recursive theory is used. When using deduced tree-structures for a given application, these tree-structures thus may have to be pre-processed so as to reduce the statistical biases.

On the other hand we could show that deduced tree-structures contain new pieces of information which may constitute additional resources for NLP applications. This could be shown by extracting frequencies of word-to-word relations and testing their correlations with an unseen corpus. When deduced frequencies are added to those obtained from real corpora, predictions become more accurate. The new information could also be attested by improving the accuracy of a parser when training deduced trees in addition to human-annotated trees. The fact that the accuracy and not only the run time may benefit from deductive EBL has been neglected in previous research. Additional experiments however failed to explain how the successful learning from deduced corpora for the parsing task took place. Neither macro learning nor changes in the attractiveness could be advanced as convincing explanation of how learning took place.

As for abduced trees, they are less biased than deduced trees. This is reflected by the frequency relations of main phrasal categories within the abduced structures and the word-to-word relations extracted from the abduced structures. Nevertheless, a strong bias remains when compared to human-annotated trees. Macro-learning may be the main reason why a parser might profit from learning abduced tree structures, besides the learning of new elementary r.

As for parsing, the overall benefit from abduced trees is close to that obtained from human-annotated trees. We have argued that the success of abduced trees is due to the relevance of the learned corpora, where relevance can be estimated by the recall of learned macros. While deduced trees are less relevant and less representative than abduced trees, human-annotated trees may be representative but are not necessarily relevant.

6 Discussion

Using different learning techniques, is seems that we trade the representativity, the relevance, the accuracy and the size of the derived corpora: Relaxing accuracy constraints may result in larger and more representative corpora with longer structures. Filtering out representative corpora from un-representative corpora reduces the size of the corpora. The advantage of human-annotated structures is that they are representative and relatively accurate. Automatic learning techniques can provide large-scale corpora, small representative corpora or corpora relevant for learning. Relevance and representativity seem to be mutually exclusive, or can be achieved jointly only with very large corpora.

These findings may be true for other learning approaches which aim at an automatic acquisition of tree-structures. If, for example, we do not care whether a parser produces a tree via deduction or abduction but simply retain those parsing trees which are above a certain numerical threshold of self-estimation of accuracy, we do not escape from the logical framework described here. With such an approach we would simply replace the qualitative explanation by a less informative quantitative self-estimation. The only degree of freedom with such an approach is to trade representative large corpora (low threshold) for accurate small corpora (high threshold). There is no way to influence the relevance.

The same may be true for the parallel run of different parsing engines on identical corpora. If different engines arrive at the same conclusion, we might assume that the results are very unlikely to be wrong and store them. However, we are relying with this approach as well as with the threshold approach or EBL on the overlap of different resources in order to make an assumption plausible. The more resources we use, the more likely the assumptions are to be correct. As a trade-off, overlaps become shorter (in terms of structure length) and less representative (e.g. mostly NPs). Again, the degree of freedom for automatic development of corpora is low. Accuracy and representativity are mutually exclusive. In addition, the so obtained corpora may be not relevant at all, as they have already been correctly processed by more than one parsing engine.

In this light, EBL seems to be the most informative approach which reports every reasoning step which has lead to the generation of a tree. Filters which are required in order to profit from automatically generated corpora can work accurately on these explanations. As shown in this paper filters may distinguish deduced trees from recursively deduced and abduced trees. With abductions, different kind of abduction steps and with them different properties of the corpora might be determined.

Although the automatically generated corpora cannot be perfect, their successful application may question human corpus annotation beyond a certain limit. As for the moment human annotators cannot be dispensed with, human resources would be used better in a framework where automatic learning approaches cooperate with human annotators. Such procedures could be:

- For every incorrectly parsed tree structure which has been corrected by a human annotator and which thus is relevant for the further improvement of the parser, a deductive learning process is started in order to increase the attractiveness of this new tree. The number of deduced trees may be limited in order not to bias the corpora.

- The annotation process starts from an abductive learning (thus assuring a high degree of relevance of the examples), submitting each abduced tree to a human annotator for control and correction. In this setting, the correctness constraints on the abduction can be relaxed so that the abduced corpora become larger and more representative. On the other hand parsing errors are less likely and less difficult to repair than those errors occurring when processing structures in the order of their occurrence.

The final conclusion may be that the distribution of labor between the computer and the human annotator in a task of corpus annotation is still far from being optimal. It is our conviction that experiments in this field will help to reduce the labor investment and error rate in what currently are the most valuable linguistic resources we have.

References

[1] AHA, D. W. Editorial- lazy learning. *Artificial Intelligence Review*, 11 (1997), 1–3.

[2] BIKEL, D. M., AND CHIANG, D. Two statistical parsing models applied the the Chinese treebank. In *Proceedings of the Second Chinese Language Processing Workshop* (Hong Kong, October 2000), M. Palmer, M. Marcus, A. Joshi, and F. Xia, Eds., Association for Computational Linguistics.

[3] BLACK, E., EUBANK, S., KASHIOKA, H., DAVID, M., GARSIDE, R., AND LEECH, G. Beyond skeleton parsing: producing a comprehensive large-scale general-English treebank with full grammatical analysis. In *COLING'96* (Copenhagen, Denmark, 1996), pp. 107–112.

[4] BOD, R. Data oriented parsing (dop). In *COLING'92* (1992).

[5] BOGUSLAVSKIJ, I., GRIGOREV, N., GRIGOREVA, S., IOMDIN, L., KREIDLIN, M., SANNIKOV, V., AND FRID, N. Annotirovannyj korpus russkix tekstov: koncepcija, instrumenty razmetki, tipy informacii. In *Proceedings of the Dialogue'2000 International Seminar in Computational Linguistics and Applications, Volume 2* (Prodvino, Russia, 2000), pp. 41–47.

[6] BRILL, E. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics* (December 1995).

[7] CHARNIAK, E. Tree-bank grammars. In *13th National Conference on Artificial Intelligence, AAAI-96* (1996), pp. 1031–1036.

[8] CHARNIAK, E. A maximum entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics* (Seattle, Washington, 2000), pp. 132–139.

[9] CHEN, K.-J., AND LIU, S.-H. Word identification for Mandarin Chinese sentences. In *COLING'92* (1992).

[10] CHEN, K.-J., LUO, C.-C., GAO, Z.-M., CHANG, M.-C., CHEN, F.-Y., AND CHEN, C.-J. The CKIP Chinese Treebank. In *Journées ATALA sur les Corpus annotés pour la syntaxe* (1999), Talana, Paris VII.

[11] CHIANG, D. Statistical parsing with automatically-extracted Tree Adjoining Grammar. In *38th Annual Meeting of the Association for Computational Linguistics* (Hong Kong, October 2000), pp. 456–463.

[12] COLLINS, M. A new statistical parser based on bigram lexical dependencies. In *34th Annual Meeting of the ACL* (1996).

[13] HAJIČOVA, E., PANEVOVA, J., AND SGALL, P. Language resources need annotations to make them really reusable: The Prague dependency treebank. In *First International Conference on Language Resources & Evaluation, Granada, Spain* (1998).

[14] HAN, C.-H. Bracketing Guidelines for the Penn Korean Treebank (draft). URL: http://www.cis.upenn.edu/xtag/korean.tag, 2000.

[15] HUANG, C.-R., CHEN, F.-Y., CHEN, K.-J., GAO, Z.-M., AND CHEN, K.-Y. Sinica treebank: Design criteria, annotation guidelines and on-line interface. In *Proceedings of the Second Chinese Language Processing Workshop* (Hong Kong, October 2000), M. Palmer, M. Marcus, A. Joshi, and F. Xia, Eds., Association for Computational Linguistics.

[16] HUANG, C.-R., AND CHEN, K.-J. A Chinese corpus for linguistics research. In *COLING'92* (Nantes, 1992).

[17] KOVARIK, J. How should a large corpus be built? - A comparative study of closure in annotated newspaper corpora from two Chinese sources. In *Proceedings of the Second Chinese Language Processing Workshop* (Hong Kong, October 2000), M. Palmer, M. Marcus, A. Joshi, and F. Xia, Eds., Association for Computational Linguistics.

[18] KUROHASHI, S., AND NAGAO, M. Building a Japanese parsed corpus while improving the parsing system. In *First International Conference on Language Resources & Evaluation, Granada, Spain* (1998), pp. 719–724.

[19] LEAKE, D. Abduction, experience, and goals: A model of everyday abductive explanation. *Journal of Experimental & Theoretical Artificial Intelligence 7* (1995), 407–428.

[20] MANNING, C. D., AND SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, London, 1999. URL http://www.sultry.arts.usyd.edu.au/cmanning/.

[21] MARCUS, M. P., SANTORINI, B., AND ANN, M. M. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics 19*, 2 (1993), 313–330.

[22] NEUMANN, G. Application of explanation-based learning for efficient processing of constraint-based grammars. In *The 10th Conference on Artificial Intelligence for Applications* (San Antonio, Texas, 1994).

[23] O'RORKE, P. Abduction and explanation-based learning: Case studies in divers domains. *Computational Intelligence 10*, 3 (1994), 295–330.

[24] PAIJMANS, H. Gravity wells of meaning. *Journal of Documentation 53* (1997), 520–36.

[25] RAYNER, M., AND SAMUELSSON, C. Corpus-based grammar specification for fast analysis. In *Spoken Language Translator: First Year Report*, SRI Technical Report CRC-043, pg. 41-54. 1994. URL http://www.cam.sri.com.

[26] ROSCH, E., AND MERVIS, C. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology 7* (1975), 573–605.

[27] SCHA, R., BOD, R., AND SIMA'AN, K. A memory-based model of syntactic analysis: Data-oriented parsing. *Journal of Experimental & Theoretical Artificial Intelligence Special issue on Memory-based Language Processing*, 11 (1999), 409–440. URL http://www2.netcetera.nl/∼iaaa/rs/jetai/jetai.html.

[28] SCHMID, H. Part-of-speech tagging with neural networks. In *15th International Conference on Computational Linguistics* (Kyoto, Japan, August 1994), pp. 172–176.

[29] SHIEBER, S. M., AND SCHABES, Y. Synchronous tree-adjoining grammars. In *COLING'90* (1990), vol. 1, pp. 1–6.

[30] SIMOV ET AL., K. Building a linguistically interpreted corpus of Bulgarian: the Bultreebank. In *Proceedings of LREC 2002* (Canary Islands, Spain, 2002).

[31] SRINIVAS, B., AND JOSHI, A. K. Some novel applications of explanation-based learning to parsing lexicalized tree-adjoining grammars. In *33th Annual Meeting of the ACL* (Cambridge, MA, 1995). cmp-lg archive 9505023.

[32] SRINIVAS, B., AND JOSHI, A. K. Supertagging: An approach to almost parsing. *Computational Linguistics 25*, 2 (1999), 237–265.

[33] STREITER, O. Corpus-based parsing and treebank development. In *ICCPOL 2001, 19th International Conference on Computer Processing of Oriental Languages* (Seoul, Korea, 2001), pp. 115–120.

[34] STREITER, O. Memory-based parsing: Enhancing recursive top-down fuzzy match with bottom-up chunking. In *ICCPOL 2001, 19th International Conference on Computer Processing of Oriental Languages* (Seoul, Korea, 2001), pp. 219–224.

[35] STREITER, O. Recursive top-down fuzzy match: New perspectives for memory-based parsing. In *PACLIC 2001, Language, Information and Computation, Proceedings of the 15th Pacific Asia Conference* (Hong Kong, 2001), pp. 345–356.

[36] STREITER, O. Abduction, induction and memorizing in corpus-based parsing. In *ESSLLI-2002 Workshop on "Machine Learning Approaches in Computational Linguistics"* (Trento, Italy, August 5-9 2002).

[37] TADEPALLI, P. A formalization of explanation-based macro-operator learning. In *IJCAI, Proceedings of the International Joint Conference of Artificial Intelligence* (Sydney, Australia, 1991), Morgan Kaufmann, pp. 616–622.

[38] TOOLE, J., POPOWICH, F., NICHOLSON, D., DAVIDE, T., AND PAUL, M. Explanation-based learning for machine translation. In *IAI Working Paper No.36. Hybrid Approaches to Machine Translation*, O. Streiter, M. Carl, and J. Haller, Eds. 2000.

[39] XUE, N., XIA, F., HUANG, S., AND KROCH, A. The bracketing guidelines for the Penn Chinese Treebank (draft II). Technical report, University of Pennsylvania, URL http://www.ldc.upenn-ctb/, 2000.