Using DOP for transformations

Andreas van Cranenburgh[*] Joost Winter[†]

June 5, 2010

Statistical Structure in Language Processing,
University of Amsterdam
Project report, first draft

Abstract

In this paper, we investigate Data-Oriented Transformation, based on the existing formalisms of Data-Oriented Parsing, synchronous grammar, and the Goodman Reduction. The goal is to define and evaluate a general formalism for learning arbitrary transformations composed of movement and insertions based on annotated exemplars. Parsing a sentence should result in a pair of trees corresponding to the original and a transformed sentence. This stochastic approach contrasts with purely rule-based formalisms such as transformational grammar, in which abstract representations are manipulated with *a priori* specified operations. The test case will be declarative versus interrogative sentences.

[*] 0440949, acranenb@science.uva.nl

[†] 9928545, jwinter@science.uva.nl

Contents

1 Introduction

Data Oriented Parsing (DOP) has proven to be a successful framework for the parsing of natural language and several other applications in computational linguistics. In this paper, we will explore options of using a DOP-based method to be able to perform frequently occurring transformations in natural language. The main example used of such transformations will be that between declarative and interrogative sentences in English, but the developed framework should be generalizable to other transformations, such as:

- Transformations between affirmative and negative sentences.
- The transformation between the 'main clause' and 'auxiliary clause' sentence order in Dutch.
- Active versus passive sentences, focus, wh-fronting, etc.

For this project, we make use of a number of existing frameworks that may prove useful to us. In particular, these include:

- Synchronous grammars, as developed by Chiang (2006).
- Data Oriented Translation framework by Poutsma (2000b;a).
- The Goodman-reduction, as developed by Goodman (2003).

In section 2, these underlying frameworks (as well as their relevance to our project) will be explained in a somewhat more detailed fashioned; then in section 3, our own approach will be described more comprehensively; in section 4, further details regarding our implementation will be given, and in section 5, the obtained results will be presented.

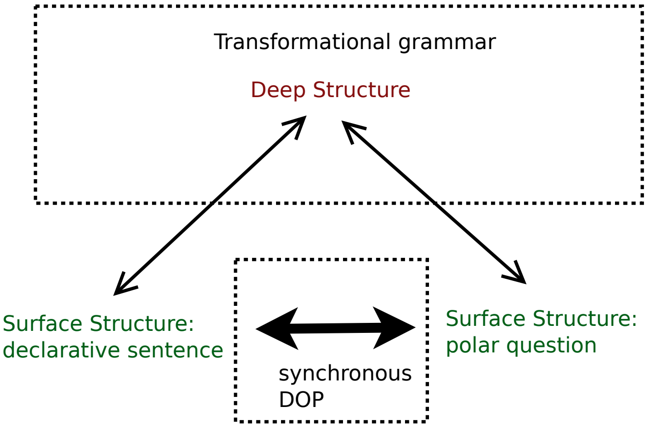

Transformational grammar
Deep Structure
Surface Structure: declarative sentence
Surface Structure: polar question
synchronous DOP

Figure 1: Overview of transformations: the orthodox approach (transformational grammar), contrasted with our data-oriented approach

2 Theoretical framework

We will use a data-oriented approach. This is opposed to the abstract, rule-based formalisms such as transformational grammar, which assume abstract structures in addition to surface structures. See figure 2 for a diagrammatic overview the relation of our project to traditional, generative approach.

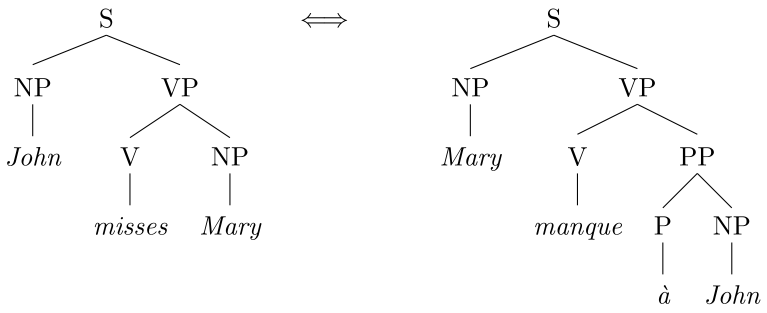
2.1 Synchronous grammars

In Chiang (2006), the concept of a synchronous grammar is introduced, as a tool to easily extend the notion of a (P)CFG into the domains of translation and transformation. The main idea, here, is that instead of single CFG-rules with a left and right hand side, every rule now has a pair of right-hand sides – one corresponding to the source language, and the other corresponding to the target language. Moreover, there has to be a bijective function between the nonterminal symbols on both right-hand sides.

Parsing and translating using synchronous grammars is just as easy as it is for regular PCFGs: to translate, we simply parse using the right-hand sides corresponding to the source language. By reading off the right-hand sides corresponding to the target language, of the rules used in the derivation, the translation is immediately obtained.

Some problems with synchronous grammars are also mentioned in Chiang (2006). One of these problems is that, in reality, tree structures often do not coincide between the original language and the translated language. For example:

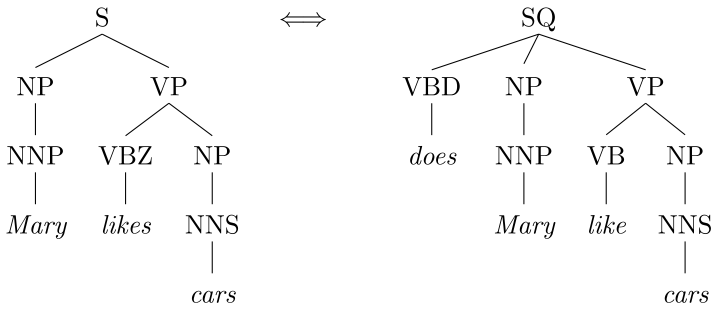
(1)



A common analysis of (1) is problematic under the original framework of synchronous grammars, because of the mismatch in the tree structures.

Similarly, when looking at transformations between declarative and interrogative sentences, and basing ourselves on standard analyses of the Stanford parser (which are based on the annotation of the Penn treebank), we encounter the following pairs of trees:

(2)



Again, we here are faced with a pair of sentences, where the tree structures are rather different from each other.

As a solution to the problem of a mismatch, the notion of flattening is named. In this case, several CFG-rules are collapsed into a single rule, so that the mismatch between the trees is resolved.

2.2 DOP for translations

In Poutsma (2000b;a), synchronous grammars are extended from plain PCFGs to the DOP framework: instead of looking at rules with two right-hand sides and a corresponding set of non-terminals, we now consider pairs subtrees with identical top nodes, and corresponding non-terminal leaf nodes.

These linked subtrees are then combined into a bag of linked subtree pairs, which can then be used as a grammar for translations and transformations. Just like in the case of synchronous PCFGs, we can simply parse new sentences using the source 'side' of the grammar, and derive the corresponding target side immediately.

Formally, in Poutsma (2000a), linked subtree pairs are defined as follows:

Definition 2.1. Given a pair of linked trees $\langle T_1, T_2 \rangle$, a *linked subtree pair* consists of two connected and linked subgraphs $\langle t_1, t_2 \rangle$ such that:

1. for every pair of linked nodes in $\langle t_1, t_2 \rangle$, it holds that
   (a) both nodes in $\langle t_1, t_2 \rangle$ have either zero daughter nodes, or
   (b) both nodes have all the daughter nodes of the corresponding node in $\langle T_1, T_2 \rangle$
2. every non-linked node in either t_1 (or t_2) has all the daughter nodes of the corresponding node in T_1 (or T_2), and
3. both t_1 and t_2 consist of more than one node.

Then, the *bag of linked subtree pairs* is defined as follows: given a corpus of linked trees C, the *bag of linked subtree pairs* of C is the bag in which linked subtree pairs occur exactly as often as they can be identified in C.

An important contrast with the earlier synchronous grammars for PCFGs, is that in the framework of Poutsma (2000a), we are not necessarily dependent on strong similarities between the tree structures on both sides (although, ultimatlely, a strong similarity may still be a prerequisite for good performance). This would make translations and transformations, such as in the examples given in section 2.1, possible.

Of course, it should be noted that, in this framework, it is unlikely that *all* subtrees on the source side are linked to subtrees on the target side: especially in the (far from unlikely) case where the total number of subtree differs, this has to be trivially false.

A problem with the DOT model, as described in Poutsma (2000a), consists of the fact that the complexity of parsing new sentences can be quite huge, as the number of subtrees can be (at worst) exponential in the number of nodes. For standard DOP-based parsing the Goodman reduction, which will be described in the next section, has been found to resolve such issues: however, because we do not have a complete set of subtrees available to use, it is impossible to use this reduction right away. This question of finding a suitable reduction for the DOT model as described in Poutsma (2000b;a), will be addressed at a later stage in this report.

2.3 Goodman reduction

A significant problem with DOP parsing, is the fact that the number of subtrees that have to be included is exponential in the size of the tree. In Goodman (2003), a solution to this problem is presented, that is able to yield equivalent parse probabilities (but not equivalent derivation probabilities) to regular DOP parsing, while limiting the number of rules to a number that is linear in the number of tree nodes.

Here, the subtrees are replaced by a set of PCFG-rules, where nodes can be either ‘addressed’ or ‘non-addressed’: in every tree, every node has a unique address, and the addressed nodes only match with that specific address, whereas the non-addressed nodes match with any node of the type.

It is proven in Goodman (2003) that the PCFG resulting from the Goodman reduction leads to equivalent parse probabilities as the underlying STSG:

Theorem 2.2. *This construction [the Goodman-reduction] produces PCFG trees homomorphic to the STSG trees with equal probability.*

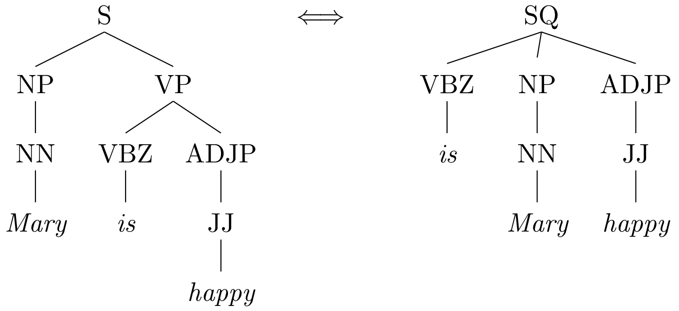
3 Approach

3.1 Definition of DOP model

We will assume a parallel corpus of pairs of trees with declarative and interrogative versions of sentences. This assumption lies on the extreme end of supervised parsing; other options are a bag of unrelated sentences marked as declarative or interrogative in their root node. Whether the assumption of a parallel corpus is cognitively plausible in terms of language acquisition is irrelevant because the point is to learn transformations from exemplars alone, without relying on built in primitives for movement and insertions, or on a common, abstract representation for sentences such as Deep Structures or Logical Form in generative grammar.

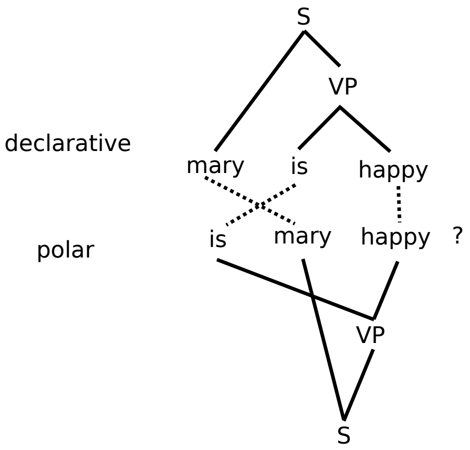
Assuming ordered pairs of trees, $\langle T_s, T_t \rangle$, may be enough to learn transformations from data alone. Naturally the word forms will often coincide, so words are implicitly linked (although this may be problematic with multiple occurrences in the same sentence). Further annotation could add indices for each constituent, but this implies labor-intensive annotation. More difficult than the linking problem is the problem of structure mismatch. Consider the following example:

(3)



In the phrase-structure annotation of the interrogative of (3) (as produced by the Stanford parser), the VP constituent has arguably become a discontinuous constituent (Harman, 1963), but due to a strong, pragmatically motivated preference for orthodox phrase-structures in computational linguistics (as well as a completely indefensible Anglocentrism), such information is sacrificed by collapsing the phrase-structure. This means that the phrase-structure no longer encodes the connection of the verb and its direct object. A more elaborate representation such as employed in the discontinuous constituent phrase-structure grammar of Harman (1963) would make it possible to streamline the annotation of sentences so that less configurational information can be preserved. Note that discontinuous constituents are only a minor step in the spectrum of constituency versus dependency. While a discontinuous constituent violates the canonical definition of a tree, it is still word order which determines constituency (e.g., in the example a fronted auxiliary verb signifies VSO order), and it is not obvious that a free word-order language could be profitably described by such a representation. When (3) is represented using discontinuous constituents, the tree could look like this:

(4)

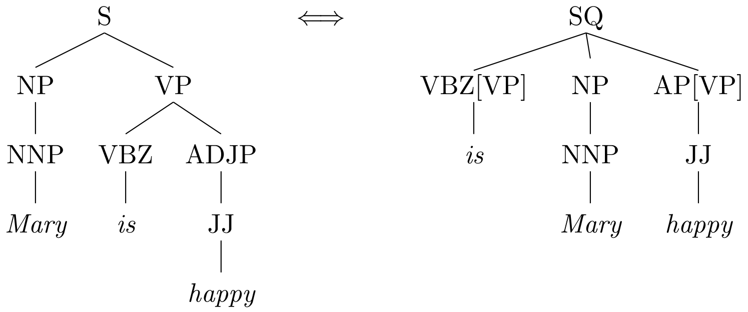


It is clear that this kind of representation provides a way to avoid structure mismatch as a result of movement completely (ie., the bag of non-terminals will be equivalent for both structures). The absence of the auxiliary 'did' in the declarative such as in (2) can be represented by trace elements; an even simpler strategy would be to allow structure mismatch as long as one of the trees is a proper subtree of the other. The reason that it may be worthwhile to avoid structure mismatch is that it maximizes the number of parallel subtrees, which minimizes the violation of the DOP hypothesis stating that all fragments should count in order to fully exploit the data at hand. It may turn out that without isomorphic phrase-structures, the generalizing power of synchronous DOP is not sufficient to handle complex, novel sentences (long constituents, relative or embedded clauses, etc.).

The same definition for subtrees and fragments can be applied to discontinuous structures, it is basically a matter of annotation and implementation to employ them, not formalism. Whether it will be necessary or worthwhile to adopt discontinuous constituents will have to be determined emprically. Initially we will attempt to use the standard PTB phrase-structure annotation and see how well it works; this has the obvious advantage of the availability of large corpora such as the WSJ, which can complement a small, hand-annotated parallel treebank.

As a compromise between the impoverished PTB annotation and the rich but non-standard discontinuous annotation, we could consider labelling POS tags with (former) parent annotations, e.g.:

(5)



As operations we will restrict ourselves to the standard substitution operation. While it may seem obvious to extend the DOP framework with movement and adjunction operations, that has the downside of having to adapt the existing, well-defined formalisms, including their probability models. On top of that we set out to induce transformations from data alone, so making such things part of the formalism means smuggling in *a priori* expectations (albeit only in a very general sense).

As estimation method we will employ the standard DOP1 estimator, but DOP* (Zollmann and Sima'an, 2005) or backoff-DOP (Sima'an and Buratto, 2003; Prescher et al., 2004) can be considered as well. Although sparsity of evidence is claimed to be an insurmountable problem for non-nativist theories by generativists (cf. MacWhinney (2004) for a discussion), for example the transformation of aux-fronting seems simple enough given elaborate information such as phrase structures; e.g. Bod (2007) shows that auxiliary fronting is learnable from positive evidence from Childes data annotated with dependency structures. For this reason it would seem that DOP1 should be adequate.

3.2 A reduction for synchronous DOP

As an instantiation of the previously mentioned assumptions, we will apply Data-Oriented Translation to transformations. For implementation efficiency we would like to use the Goodman reduction.

We are now faced with the situation, that we cannot simply apply the Goodman reduction, reducing subtree pairs to PCFG rules, because of the reason that not every subtree will always be a part of a subtree pair. In particular, the PCFG-rules themselves, the 'building blocks' of the Goodman reduction, may not be a part of a subtree pair. This leads us to wonder whether it is possible to, in some way or another, still perform a Goodman-like reduction on the bag of linked subtree pairs.

It turns out that, indeed, a reduction is possible over the bag of linked subtree pairs. For this, consider the notion of a *minimal linked subtree pair*:

Definition 3.1. For a pair of linked trees $\langle T_1, T_2 \rangle$, a *minimal linked subtree pair* is a linked subtree pair $\langle t_1, t_2 \rangle$, s.t. $t_1 \subseteq T_1$, $t_2 \subseteq T_2$, such that there is no other linked pair $\langle s_1, s_2\rangle$ where $s_1 \subsetneq t_1$ and $s_2 \subsetneq t_2$.

Theorem 3.2. *Every pair of linked trees can be reconstructed from the minimal linked subtree pairs derived from it.*

Proof. Assume that there is a pair of linked trees $\langle T_1, T_2\rangle$ for which this is not possible, given a certain bag of linked subtree pairs. Then there has to be some linked pair of subtrees $\langle t_1, t_2\rangle$ for which:

1. It is not possible to construct $\langle t_1, t_2\rangle$ from minimal linked subtree pairs.
2. All pairs of subtrees $\langle s_1, s_2\rangle \subsetneq \langle t_1, t_2\rangle$ can be constructed from minimal subtree linked pairs.

It is impossible that this pair of subtrees is itself minimal: if it were, it could trivially be constructed from linked minimal subtree pairs. So, for this case to hold, the linked subtree pair $\langle t_1, t_2\rangle$ must have a proper subset $\langle s_1, s_2\rangle$.

...(incomplete but I know how to finish)

□

In other words: just like the PCFG rules can be used as the 'building blocks' for the Goodman reduction, the minimal linked subtree pairs as defined above can be used as the 'building blocks' for a similar reduction. It should be noted that, for all means and purposes, such minimal subtrees can be worked with, as if they were simple CFG-rules, that just happen to have an additional internal structure (which is ignored by the parser).

Like in the Goodman-reduction, we replace every elementary tree (i.e. every minimal linked subtree pair) with $2^{(k+1)}$ new rules, where k is the number of nonterminals occurring as leaf nodes (which has to be equal on the left and right hand sides). For the root node, and for every nonterminal node, either an 'addressed' version or a 'non-addressed' version can occur.

Theorem 3.3. *A 'Goodman-like' reduction based on minimal linked subtree pairs leads to equivalent parse probabilities (but not equivalent derivation probabilities) as the original synchronous DOP model.*

4 Implementation

TBD.

5 Evaluation

Standard PARSEVAL measures (with `EVALB`), transformations in both directions. Take (part of) WSJ treebank, hand annotate corresponding transformed sentences.

TBD.

6 Conclusion

TBD.

References

Bod, R. (2007). A linguistic investigation into unsupervised DOP. *ACL 2007*, page 1. Available from: http://acl.ldc.upenn.edu/W/W07/W07-0601.pdf.

Chiang, D. (2006). An introduction to synchronous grammars. *Tutorial on ACL-06*. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.3212&rep=rep1&type=pdf.

Goodman, J. (2003). Efficient parsing of DOP with PCFG-reductions. *Bod et al.(Bod et al., 2003)*. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.6394.

Harman, G. (1963). Generative grammars without transformation rules: a defense of phrase structure. *Language*, 39(4):597–616. Available from: http://www.jstor.org/stable/411954.

MacWhinney, B. (2004). A multiple process solution to the logical problem of language acquisition. *Journal of Child Language*, 31(04):883–914. Available from: http://journals.cambridge.org/abstract_S0305000904006336.

Poutsma, A. (2000a). Data-oriented translation. In *18th COLING*, pages 635–641. Available from: http://acl.ldc.upenn.edu/C/c00/c00-2092.pdf.

Poutsma, A. (2000b). Data-Oriented Translation: using the Data-Oriented Parsing framework for machine translation. *Master's thesis, University of Amsterdam, Amsterdam, the Netherlands*. Available from: http://www.xs4all.nl/~ajwp/main.ps.gz.

Prescher, D., Scha, R., Sima'an, K., and Zollmann, A. (2004). On the statistical consistency of DOP estimators. *Proceedings CLIN 2004*. Available from: http://dare.uva.nl/record/190343.

Sima'an, K. and Buratto, L. (2003). Backoff parameter estimation for the DOP model. *Machine Learning: ECML 2003*, pages 373–384. Available from: http://dare.uva.nl/record/126078.

Zollmann, A. and Sima'an, K. (2005). A consistent and efficient estimator for Data-Oriented Parsing. *Journal of Automata Languages and Combinatorics*, 10(2/3):367. Available from: http://staff.science.uva.nl/~simaan/D-Papers/JALCsubmit.pdf.