Getting Rid Of Numbers

DOP without Probability Theory?

Malvin Gattinger, MoL student (# 10407952)

2012/13 II a: Cognitive Models for Language and Music

Probabilistic context free grammars (PCFGs) and similar models make use of probability theory and therefore depend on numeric calculations. In particular, the Data-Oriented-Parsing (DOP) model from [1] uses the relative probability of subtrees to determine preferred derivations and thereby preferred trees for given sentences.

This motivates criticism of its cognitive realism from both philosophical and mathematical-computational perspectives. The question arises whether such models can be reformulated without the usage of numeric probabilities.

Inspired by a general research question from [7] we will try to capture the exact role of probabilities in DOP and investigate the question whether we can replace its numeric calculations with other, non-numeric structures. We show the negative result that simple linear orderings will not suffice as a replacement. Trying to "repair" their failure we introduce and investigate the idea of meta-orderings and state further questions they create.

Contents

1 Data-Oriented Parsing and probabilities

1.1 Basics and variations of DOP

The Data-Oriented-Parsing Model originally presented in [1] provides a cognitive model of language learning, understanding and usage. While it can in principal be extended to any structured or annotated form of language samples, the original DOP which we will be concerned with works on simple trees.

Given a treebank, DOP generates all contiguous subtrees[1] and assigns probabilities to each of them. The probability of a tree is defined as the ratio between its number of occurrences in the treebank and the total number of subtrees starting with a node of the same category.

Using these subtrees as building blocks, we can give derivations for trees for (lots of) new sentences which were not part of the original treebank. Note that multiple derivations for the same tree are possible and we can have multiple trees for the same sentence.

Finally, the probability distribution on subtrees allows us to define probabilities of derivations, trees and sentences: First, the probability of a derivation is defined as the product of the probabilities of its parts. Second, the probability of a tree is defined as the sum of the probabilities of all its derivations. Third, the probability of a sentence is defined as the sum of the probabilities of all its trees.

The model can thus select the preferred derivation for a given tree, the preferred tree for a given sentence and the most probable sentence given a collection. Note that these selections do not have to be unique as the probability of different derivations might be the same. Fortunately, this is almost never the case as long as the treebank is big enough.

There are many variations of DOP, one of which is the unsupervised U-DOP presented in [4]. It does not use an annotated treebank but starts with equally considering all non-categorical binary trees for all sentences. The discussion below applies to U-DOP in so far as the probabilities and therefore also the orderings will always concern all subtrees and not only the ones with a certain category at the root.

Most DOP variants select the derivation with the highest probability as the preferred one. While other criteria have also been considered, they do not perform as well and still rely on probabilities as a fall-back option: While their main criteria can be for example the derivation length, in case of a draw they still select the more probable derivation among the shortest ones.

Furthermore, DOP can work on different kinds of structures which need not be tree-like, e.g. mere sequences. For simplicity reasons we will now restrict ourselves to variants of DOP working with binary trees using only the categories S, NP, VP, V and AP.

[1] Note that in DOP subtrees are generated by both picking non-terminals as the new root and cutting off one or more leaves or whole branches at the same time. For example, `(NP(VP)(NP bees))` is a subtree of `(S(NP(VP kissing)(NP bees))(VP(V fly)))`.

1.2 The role of probabilities in DOP

It is already clear from the introduction that DOP makes heavy use of numeric calculations with probabilities. Note that using multiplication on the rational interval $(0, 1]$ might seem trivial but is crucial: A distribution over subtrees can only induce one over derivations because the product of probability distributions always yields another. In particular this allows us to compare the probability of derivations of different and arbitrary length.

On the other hand, one can argue that the usage of numeric values is not essential to DOP: All the numbers are doing and all we actually need to parse new sentences, i.e. select preferred derivations, are orderings on subtrees, derivations and sentences. It does not seem necessary or fundamental to DOP that these three orderings are induced by a (even one and the same) probability distribution. While the preference towards certain constructions and therefore meaning is represented using numbers, we want to point out that it need not be of a numeric value itself.

Even though actual implementations might be different and only do the actually needed calculations, a complete DOP model generated from a certain treebank can "forget" about the numbers and still provide a parser: After all subtrees are generated, their frequencies are multiplied to induce an ordering on derivations of new, unknown trees. Once this ordering is computed for all trees up to a certain length, the concrete numbers do not matter any more, because the output of the model, namely the three orderings is not numerical.

From this perspective, the usage of probabilities in DOP seems is merely an intermediate step. Thus we will explore whether one could go one step back and replace this intermediate numeric step by purely structural and logical reasoning. Our aim is to find another, preferably non-numeric way from a given treebank to an ordering of subtrees, an ordering of derivations and an ordering of sentences.

Our first candidate (discussed in section 2.1 and 2.2) for such a replacement is still very similar to the original: We presuppose a linear, though non-numeric ordering on the subtrees and will try to derive one on derivations from this.

1.3 Motivations to get rid of numbers

Besides the described "intermediate step" perspective, there are other arguments to investigate whether DOP depends on probability theory. We briefly discuss a general inspiration, the criticism of cognitive realism and a mathematical argument.

A general research question

This paper was heavily inspired by a general research question raised by Johan van Benthem in [7]: When and where is probabilistic reasoning really essential? Where and how could it be replaced or simulated by purely logic reasoning?

While the question originally targets dynamic epistemic logic and update semantics, it also wants to intensify the dialogue between logic and probability theory in general. Hence

PCFGs and DOP models in particular lend themselves as objects for the (seemingly?) naive but also surprisingly neglected "Do we have to use numbers here?" question.

Cognitive realism

A major reproval against DOP concerns its cognitive realism[2]: The (linear) speed and intuitive way we form and parse new sentences speaks more against than for an underlying complicated numerical process involving the comparison of many different constructions and meanings in parallel. Such doubts are of the form and content "Do we really remember every single construction and their frequency?" and the more plausible answer is negative.

A mathematical argument

From a mathematical perspective, probability theory is a two-sided medal: It allows for an easy comparison of probabilities of arbitrary complex events, but can also hide reasoning behind numbers that might as well be done without them. In our case we have an in principle finite and discrete set of events, but for convenience we use the usual set of rational numbers to reason about them. Hence we should explore if a mathematically simpler concept of e.g. orderings would suffice.

It should be stressed here that this and the previous argument go against too many numeric calculations in general, not only the fact that DOP usually involves very small fractions. The latter is a more phenomenological concern and a matter of implementation. It can be responded to by using negative logarithms instead of the original fractions to represent probabilities.

A computational argument

Another motivation for investigating non-numeric adoptions of the DOP-model can be the search for more efficient algorithms. Current implementations have to make use of Monte Carlo techniques or PCFG-reductions to be efficient. Therefore it is of interest if the complexity of implementing DOP is due to the calculations involved or the logical/-combinatorial recursion explosion. Though, this argument might also turn out to be a counterargument, depending on the complexity of computing the alternative structure.

1.4 Existing work: Probabilities and PCFGs

Concerning expressiveness of PCFGs in general, the following has been shown:

> "Probabilities cannot be mimicked by rules, i.e., their use is fundamental from a formal language perspective: whenever used as a filtering mechanism, probabilities can define a set of trees that can not captured by CFGs." [5, p. 3]

[2]By "realism" we do not mean a philosophical school but the property of a theory of being cognitively realistic and plausible.

Though [5] and [6] are also investigating what exactly the contribution of probabilities to CFGs is, they do so via the tree language defined by them. Using an inherently ambiguous language they show that a CFG without any additional structure will not be able to define the same tree language as a PCFG. But they do not investigate if there are other structures besides probability distributions which one could add to a CFG to define the same language. Therefore their result is not a general negative answer to our question.

Still, the (also not general in this sense) proof we give in section 2.2 is similar to the one in [5], but more concretely arguing about a DOP model and not relying on previous results about inherently ambiguous language.

2 Simple and Meta-Orderings

2.1 The idea

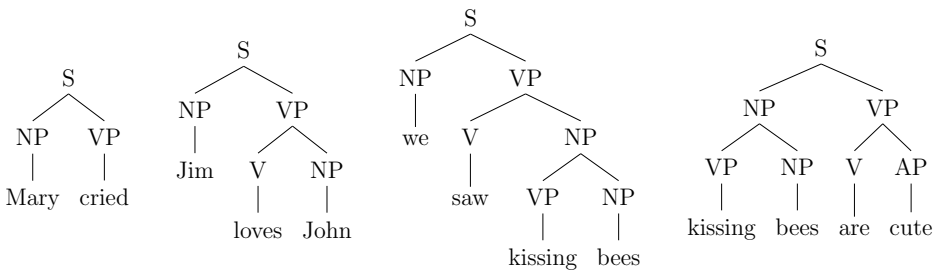
The intuition behind trying to use a single linear ordering of all subtrees instead of probabilities is the following: If we can sort all our known constructions (i.e. subtrees) by our preference, then this ordering should be enough to also decide between derivations.

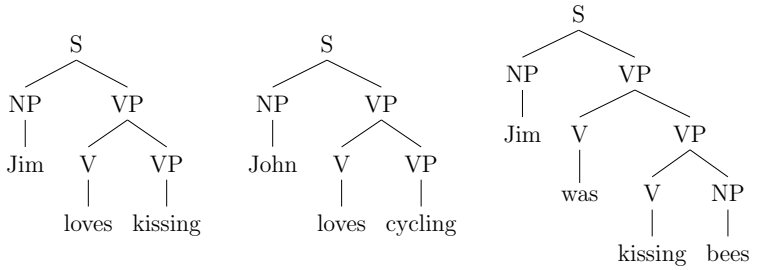
Suppose we have a linear ordering on the set of all subtrees of trees in our treebank. For example this linear ordering can be obtained from the number of occurrences or even from the probabilities, but its origin is not at stake right now. We merely think of it as some representation of all conscious and unconscious thoughts of the form "Someone would rather say A than B". We do not know absolutely how frequent a certain construction is but we know that some are more frequent than others. Our question is now: Can such a linear ordering uniquely select a preferred derivation "as good as" the usual probability distributions can?

The next section provides a proof for the negative answer by giving a counterexample. We generate a DOP model with a linear ordering of subtrees that does not induce one on the derivations which is equivalent to the one obtained from probabilities.

2.2 Simple Orderings do not suffice

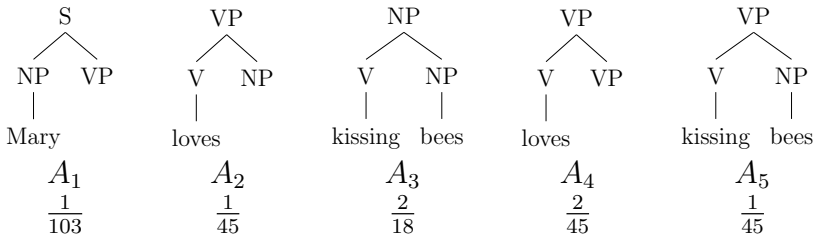
Consider the following example, a DOP model generated from a small example treebank containing the following seven trees:





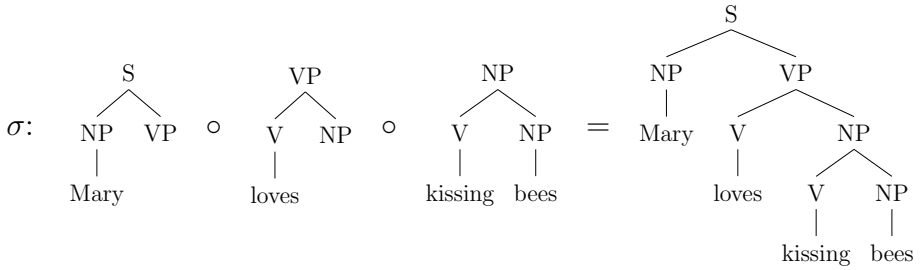
Remember that the complete DOP model consists of all subtrees of these trees. A strong motivation for this is that as shown in [3] any systematic restriction on subtree length does not increase the performance of a DOP model.

In total the generated DOP contains 174 subtrees with the following root node categories: 103 S, 18 NP, 45 VP, 7 V and 1 AP. Among these we have the following subtrees with the corresponding probabilities:[3]



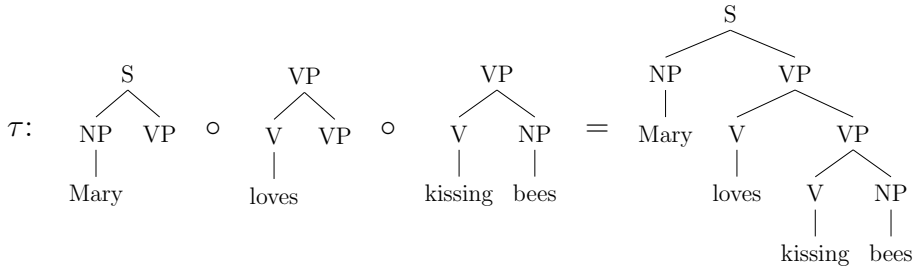
Note that $P(A_4) > P(A_2)$ and $P(A_3) > P(A_5)$. Hence this corpus reflects the following two assumptions about "loves" and "kissing bees": First, the VP "loves" is more often used to describe an agents preference to perform a certain action than her affection to something or someone. Second, "kissing bees" occurs more often as an NP referring to bees kissing each other than as a VP denoting the activity of kissing one or multiple bees.[4]

Let σ and τ be the derivations $A_1 \circ A_2 \circ A_3$ and $A_1 \circ A_4 \circ A_5$ respectively, generating two different trees for the same sentence "Mary loves kissing bees":



[3]The subtrees were generated using python and nltk.tree from [8]. See Appendix A for the source.

[4]Whether these assumptions about the actual usage of these words are realistic or not is not important - the concrete example does not matter to what we are proving here.



To decide which derivation and thus which tree is more probable according to the DOP model, we multiply the probabilities of each subtree:

$$P(\sigma) = P(A_1) \times P(A_2) \times P(A_3) = \frac{1}{103} \times \frac{1}{45} \times \frac{2}{18} = \frac{2}{83430}$$

$$P(\tau) = P(A_1) \times P(A_4) \times P(A_5) = \frac{1}{103} \times \frac{2}{45} \times \frac{1}{45} = \frac{2}{208575}$$

Hence σ is preferred. But what if instead of probabilities one would only be given the linear ordering $<$ of the used subtrees? Then we would not have access to the numeric values but still know which subtrees are more preferred than others.[5]

The fact that using A_4 is "better" than using A_2 might make us prefer τ, but A_3A_5 is an argument in favour of σ. Therefore this simple linear ordering will not contain enough information to decide which of the two derivations should be preferred.

This tells us something about the way in which DOP chooses a derivation: Not only does it matter which rules are more probable but also which of these differences in probability / relative number of occurrences are bigger than others.

In general this problem will occur if we have a situation like this:

$$\begin{aligned} \sigma = &\ \ldots \circ A_1 \circ A_2 \circ \ldots \\ \tau = &\ \ldots \circ A_3 \circ A_4 \circ \ldots \end{aligned}$$

where $A_1 < A_3$ but also $A_4 < A_2$.

2.3 Meta-orderings

If one does not completely abandon the idea of using linear orderings after their failure in the last section, the following idea comes to mind in order to "repair" them: Suppose we had a meta-ordering which tells us which of the statements $A_1 < A_3$ or $A_4 < A2$ is "more important" and outweighs the other. More formally we would like to have an ordering of the cartesian product of all subtrees, i.e. a structure $(S \times S, \prec)$ such that whenever $(A_1, A_3) \prec (A_4, A_2)$ holds, we will rather obey the first preference when selecting a derivation.

[5] Only the phrase "more probable" and notation like $P(A_1) < P(A_2)$ should be read to mean rankings of (or induced by) probabilities. For the linear ordering $<$ which does not mean to be of some lower numeric probability value, but merely being preferred because of previous language experience we write $A_1 < A_2$.

As with linear orders before, we can connect this idea to an intuition: We do not only have a preference towards certain constructions but some of these preferences are much stronger than others. In our example above this could be the fact that the difference between A_2 and A_4 is smaller than the one between A_3 and A_5. Formally we would have $(A_2, A_4) \prec (A_3, A_4)$. A parser could then use this information without referring to numbers and select the derivation that uses A_4 instead of A_3, ignoring that it uses A_4 instead of A_2 which usually should not happen. Therefore a meta-ordering would be enough to "repair" the previous example.

But problems are coming: To show that such meta-orderings are sufficient we would have to show that a linear ordering and a meta-ordering on top of it can uniquely select a preferred derivation - again "as good as" the usual probability distributions. A natural idea to prove the simulation result "Meta-orderings can be as precise as probabilities." would be to read the ordering and the meta-ordering from the probabilities and let $(A_1, A_2) \prec (A_3, A_4)$ iff $|P(A_2) - P(A_1)| < |P(A_4) - P(A_3)|$. Unfortunately, the following example shows that this is not a necessary and sufficient criteria.

Suppose we have four subtrees such that $P(A_1) = \frac{7}{10}$, $P(A_2) = \frac{9}{10}$, $P(A_3) = \frac{1}{10}$ and $P(A_4) = \frac{2}{10}$. Consider the two derivations $\sigma = A_1 \circ A_4$, $\tau = A_2 \circ A_3$.
In the regular DOP model we have $P(\sigma) = P(A_1) \times P(A_4) = \frac{14}{100}$ and $P(\tau) = P(A_2) \times P(A_3) = \frac{9}{100}$. and therefore a preference for τ. But by the suggested definition in the previous paragraph we would have $(A_1, A_2) \prec (A_3, A_4)$ and therefore our hypothetical model based on a meta-ordering would prefer σ.

3 Conclusions and further questions

We discussed the role of probabilities in the DOP model. Taking different arguments against numeric calculations seriously we tried to simulate the output of a DOP model with simple orderings of the subtrees and showed that this is not possible in general. Furthermore, meta-orderings do not provide a way out if we use a natural definition, i.e. read them from probabilities. For now it seems impossible to repair them without adding yet another structure, maybe ad infinitum.

Still, the last example does not provide a general negative result because it only shows that one proposed definition will not be sufficient and necessary. Hence, meta-orderings might still be a useful answer to the criticism aiming at cognitive realism: We do not store every construction and do not have perfect recall. But every construction influences our both conscious and unconscious ordering of preferences to use them.

We will end by stating some further questions for further research:

- There always is a correct meta-ordering, but can we learn one from a treebank without implicitly building a probabilistic DOP model?
- Are DOPs with meta-orderings more or less expressive than probabilistic ones? Aiming at the completeness discussed in [5], can we enforce $<$ and $\prec$ to be strict linear orders?

- Could it be that to order arbitrary treebanks, their generated subtrees and all derivations we always have to use a structure which is isomorphic to the rationals? Note that to show this we only have to show that we need a dense linear ordering as it is a basic result from model theory that the corresponding theory is ω-categorical.
- If not, are there other structures than orderings which could replace the probability distributions? Think of lattices, fields, groups, etc.
- One could still hope that the counterexamples we discussed do not occur too often when using real corpus data. To check this we should try to implement a "probably correct" DOP variation which uses orderings instead of probabilities. Could it be faster than current implementations?
- Finally, in order to contributing back to the general research question from [7]: Could the discussion of orderings replacing probabilities be of more help for other theories using probabilities? Or are the difficulties when one wants to get rid of numbers inherent to them, not to DOP?

Wordcount

Total number of words in text: 2849

References

[1] Rens Bod: *A computational model of language performance: Data Oriented Parsing.* In: *Proceedings of COLING-92.* Nantes, France 1992.
http://dx.doi.org/10.3115/993079.993082

[2] Rens Bod, R. Scha and K. Sima'an (eds.): *Data-Oriented Parsing.* CSLI Publications, The University of Chicago Press. 2002.

[3] Rens Bod: *What is the minimal set of fragments that achieves maximal parse accuracy?.* In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics.* pp. 66–73. Toulouse, France 2001.
http://dx.doi.org/10.3115/1073012.1073022

[4] Rens Bod: *An all-subtrees approach to unsupervised parsing.* In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.* pp- 865-872. Stroudsburg, PA, USA, 2006.
http://dx.doi.org/10.3115/1220175.1220284

[5] Gabriel Infante-Lopez, Maarten de Rijke: *A Note on the Expressive Power of Probabilistic Context Free Grammars.* In: Journal of Logic, Language and Information, Volume 15, Number 3, p.219-231 (2006)
http://dx.doi.org/10.1007/s10849-005-9002-x

An older version of the same article seems to be the following unpublished research report.

[6] Gabriel Infante-Lopez, Maarten de Rijke: *Can Probabilities Be Mimicked by Rules?* (submitted) January 20, 2004.
http://staff.science.uva.nl/~infante/papers/infante-lopez-mimicked-prob.pdf

[7] Johan van Benthem: *Interfacing Logical and Probabilistic Update.* ILLC, LoLaCo research talk, 10 December 2012. Revised at the ILLC colloquium,
http://staff.science.uva.nl/~ulle/teaching/lolaco/2012/slides/vanbenthem.pdf

[8] Steven Bird, Ewan Klein, Edward Loper: *Natural Language Processing with Python.* O'Reilly Media Inc. 2009.
http://nltk.org/book

Appendix A: trees.py

```
from nltk.tree import Tree
import itertools

tree = {}
tree[0] = Tree("(S (NP Mary) (VP cried) )")
tree[1] = Tree("(S (NP Jim) (VP (V loves) (NP John)))")
tree[2] = Tree("(S (NP we) (VP (V saw) (NP (VP kissing) (NP bees))))")
tree[3] = Tree("(S (NP (VP kissing) (NP bees)) (VP (V are) (AP cute)))")
tree[4] = Tree("(S (NP Jim) (VP (V loves) (VP kissing)))")
tree[5] = Tree("(S (NP John) (VP (V loves) (VP cycling)))")
tree[6] = Tree("(S (NP Jim) (VP (V was) (VP (V kissing) (NP bees))))")

def cutpoints(poslist):
    """
    given a list of treepositions, return the subset of cutpoints
    """
    cps=[]
    for pos in poslist:
        hasdesc=0
        if len(pos)>0: # do not cut at the root
            for k in range(2):
                if pos+(k,) in poslist:
                    hasdesc=1
            if hasdesc: # there is something to cut
                cps.append(pos)
    return cps

for i in range(len(tree)):
  t=tree[i]
  for s in t.subtrees():
    if s.height() < 3: # nothing to cut
        print s
    else:
        allpos = s.treepositions("preorder")
        cps=cutpoints(allpos)
        truthtable = itertools.product([0,1], repeat=len(cps))
        for line in truthtable:
            news = s.copy(True)
            addthis=1
            for i, cell in enumerate(line):
                if cell and cps[i]:
                    #print "cutting", news, "at", cps[i], "where we have
                       ", s[cps[i]]
                    try:
                        news[cps[i]]=s[cps[i]].node
                    except Exception:
                        #print "this cut-combination is nonsense"
                        addthis=0
            if (addthis): print news
```