Treebank Translation for Cross-Lingual Parser Induction

Jörg Tiedemann, Joakim Nivre
Uppsala University
name.surname@lingfil.uu.se

Željko Agić
University of Potsdam
zagic@uni-potsdam.de

Introduction

For **languages without any treebanks**, data-driven syntactic dependency parsing is tackled by annotation projection, model transfer and unsupervised approaches.

Here, we explore **treebank translation** as a hybrid approach. We use parallel corpora to build statistical machine translation models and translate the source language treebanks. We then project the annotations, train the parsers on the synthetic treebanks and use them in parsing.

Delexicalized and lexicalized models are tested. We compare them to the delexicalized baseline following McDonald et al. (2013).

Experiment

We use standard components and default parameters for SMT and parsing: Moses, MaltParser and MaltOptimizer. Europarl is used in building the SMT models.

Three modes for SMT are used: **dictionary lookup, word to word translation with word reordering, and full phrase-based SMT**.

We test the approach on Google Universal Treebanks as its annotation enables label projection and reliable evaluation across languages.

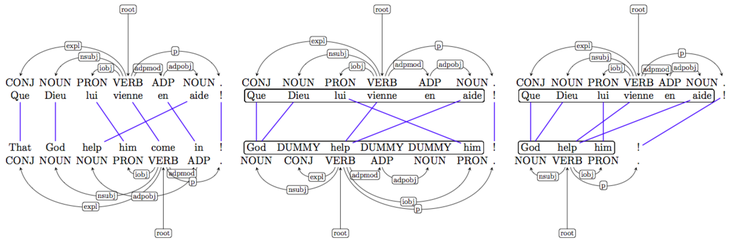
We consistently observe **substantial improvements** in LAS. Word to word translation is the top performer.

Input: source tree S, target sentence T, word alignment A, phrase segmentation P
Output: syntactic heads head[], word attributes attr[]

```
treeSize = max_distance_to_root(S) ;
attr = [] ;
head = [] ;
for t ∈ T do
    if is_unaligned_trg(t,A) then
        for t' ∈ in_trg_phrase(t,P) do
            [s_x,..,s_y] = aligned_to(t') ;
            ŝ = find_highest([s_x,..,s_y],S) ;
            t̂ = find_aligned(ŝ,S,T,A) ;
            attr[t] = DUMMY ;
            head[t] = t̂ ;
        end
    else
        [s_x,..,s_y] = aligned_to(t) ;
        s = find_highest([s_x,..,s_y],S) ;
        attr[t] = attr(s) ;
        ŝ = head_of(s,S) ;
        t̂ = find_aligned(ŝ,S,T,A) ;
        if t̂ == t then
            [s_x,..,s_y] = in_src_phrase(s,P) ;
            s* = find_highest([s_x,..,s_y],S) ;
            ŝ = head_of(s*,S) ;
            t̂ = find_aligned(ŝ,S,T,A) ;
            head[t] = t̂ ;
        end
    end
end
```

Input: node s, source tree S with root ROOT, target sentence T, word alignment A
Output: node t*

```
if s == ROOT then
    return ROOT ;
end
while is_unaligned_src(s,A) do
    s = head_of(s,S) ;
    if s == ROOT then
        return ROOT ;
    end
end
p = 0 ;
t* = undef ;
for t' ∈ aligned(s,A) do
    if position(t',T) > p then
        t* = t' ;
        p = position(t',T) ;
    end
end
return t* ;
```



Projection algorithm

Draws from the work of Hwa et al. (2005), but exploits alignment information from SMT and dependency tree properties to implement heuristics for **avoiding the introduction of dummy nodes**.

In the dictionary lookup approach, the annotation is simply copied. In word to word translation, only the word ordering changes, influencing projectivity. Only phrase-based SMT requires heuristics for handling the **many-to-many alignments**.

Ongoing work

Our projection algorithm currently introduces a lot of non-projectivity. Together with SMT quality, this most likely accounts for the overall results. We are working on better projection heuristics and better SMT by introducing tree constraints.

There is a detailed comparison of our projection and that of Hwa et al. (2005) by Tiedemann (2014).

	DELEX BASELINE					LOOKUP					WORD TO WORD					PHRASE-BASED				
	DE	EN	ES	FR	SV	DE	EN	ES	FR	SV	DE	EN	ES	FR	SV	DE	EN	ES	FR	SV
DE	62.71	43.20	46.09	46.09	50.64	-	48.63	52.66	52.06	**58.78**	-	51.86	55.90	57.77	61.65	-	50.89	52.54	54.99	59.46
EN	46.62	77.66	55.65	56.46	**57.68**	48.59	-	57.79	57.80	52.21	**53.80**	-	60.76	63.32	**62.93**	**53.71**	-	60.70	62.89	**64.01**
ES	44.03	46.73	68.21	**57.91**	53.82	47.36	49.13	-	**62.24**	57.50	49.94	49.93	-	**65.60**	59.22	49.59	48.35	-	**64.88**	58.99
FR	43.91	46.75	**59.65**	67.51	52.01	47.57	**54.06**	**66.31**	-	57.73	52.07	**54.44**	**65.63**	-	57.67	51.83	**53.81**	**65.55**	-	
SV	**50.69**	**49.13**	53.62	51.97	70.22	**51.88**	48.84	54.74	52.95	-	53.18	50.91	60.82	59.14	-	53.22	49.06	58.41	58.04	-