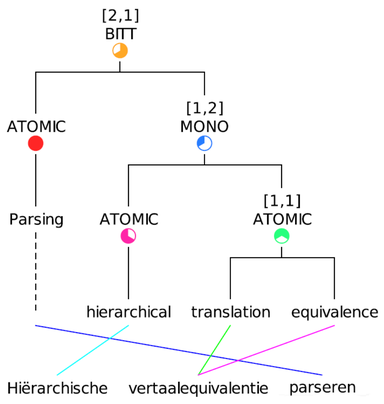

Visualization, Search and Analysis of Hierarchical Translation Equivalence

{Gideon Maillette de Buy Wenniger, Khalil Sima'an }
gemdbw@gmail.com k.simaan@uva.nl

SLPL LAB
Institute for Logic, Language and Computation
University of Amsterdam
NWO Netherlands Organisation for Scientific Research

Problem

Empirical properties of hierarchical translation equivalence as induced by word alignments are poorly known and hard to investigate due to missing representations and tools. Visualizing word level alignments gives an idea about structure but leaves hierarchical translation equivalence relations implicit. Just extracting and visualizing all phrase pairs induced by a word alignment without specifying the relations is similarly incomplete, as important information about the reordering taking place is lost. Needed is a representation of hierarchical translation equivalence that compactly represents all translation equivalents and their relations and a tool to visualize and analyze this representation.

Contributions

Our tool builds and visualizes a complete and exact representation of hierarchical translation equivalence as induced by word alignments. Translation equivalence relations with particular properties from real data can be searched and visualized. Various corpus level properties of hierarchical alignment complexity can be computed, giving more global information about the nature of translation equivalence relations for a particular language pair.

Example of HAT

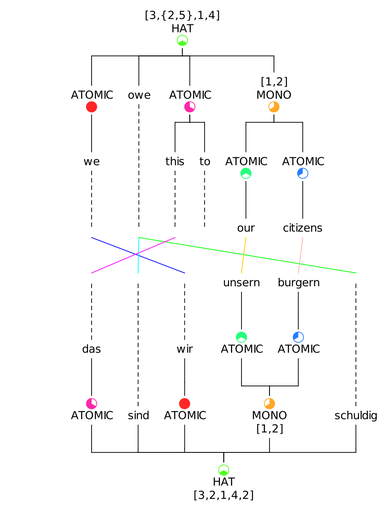

Figure 1: The visualization, filling and color of the round nodes indicates the equivalence between the top source-to-target HAT and the mirrored target-to-source HAT displayed below it. Labels, such as {3,{2,5},1,4} at the top node, denote permutation-set reordering operations at nodes.The labels *ATOMIC*, *MONO* and *HAT* in the HAT visualization indicate broad complexity categories for reordering.

Visualization

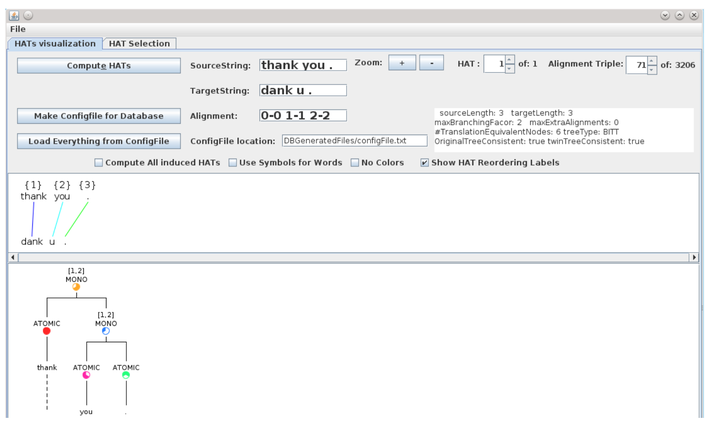

Figure 2: The HATs visualization window.

Copera of aligned sentence pairs can be browsed and HATs generated on the fly, and custom examples can be specified directly using the the input fields.

Search

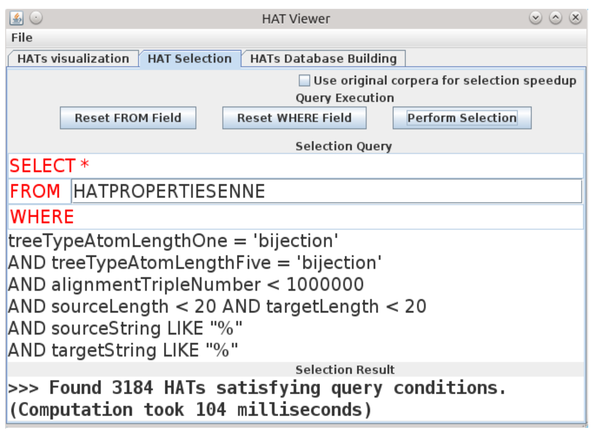

Figure 3: The database selection window of the HATs Visualization tool.

Search of HATs with certain properties is implemented as database selection. Figure 3 above shows how HATs can be selected to be of certain type (*treeTypeAtomLengthX*), be of a certain length (*sourceLength / targetLenght*) on the source and/or target and contain certain words (*targetString LIKE / sourceString LIKE*).

Method

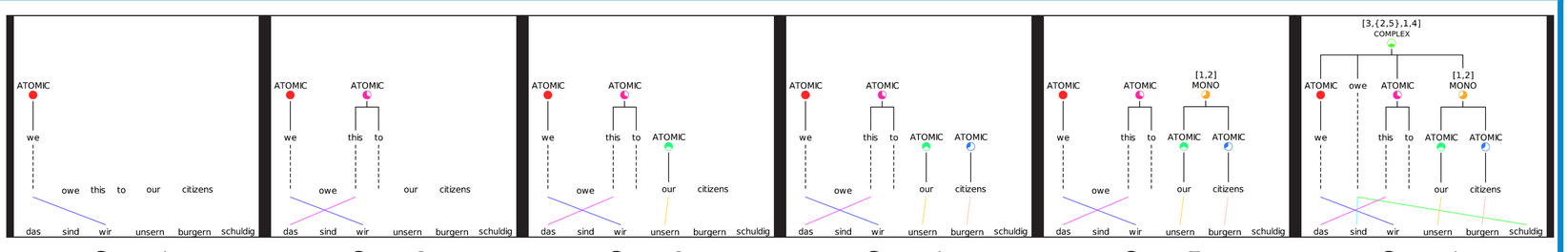

Figure 4: The stepwise composition of the example HAT in Figure 1

The desired efficient and compact representation of hierarchical translation equivalence can be achieved by **Hierarchical Alignment Trees (HATs)**[1]. **HATs** [Sima'an and Maillette de Buy Wenniger, 2013] are recursive synchronous tree pairs with nodes corresponding to phrase pairs induced by word alignments, structured to form a minimally branching factorization of these phrase pairs. They extend **Normalized Decomposition Trees (NDTs)** [Zhang et al., 2008] by providing explicit labels for the type of reordering occurring at the nodes, as well as maintaining the internal word alignments for atomic (non-decomposable) phrase pairs.

Atomic phrase pairs are phrase pairs that do not subsume smaller phrase pairs. Starting from such atomic phrase pairs larger phrase pairs are recursively built by combining a minimal number of smaller subsumed phrase pairs into larger units. Figure 4 above gives an illustration of this process, where in Step 5 the simple atomic phrases of *our* | *unsern* and *burgern* | *citizens* induced by the word alignments are composed into a bigger monotone unit.

Alignments are discontiguous when no neat phrase-based factorization into parts is possible (see the top node in Figure 1 / Figure 4 - Step 6 for an example). In such cases first the spans corresponding to proper subsumed phrase pairs are added as normal child nodes below the new phrase pair and finally the discontiguous parts are added directly as terminal productions/children below it as well. While building HATs in this recursive way, every[2] node is labeled with a *set-permutation* label. Given the local alignment at a node (phrase pair), this label specifies the relative mapping occurring directly below the node. This relative mapping is specified as an ordered list of sets of relative target mapping positions, one such set of positions for each relative position in the source phrase.[3] In the case of bijective mappings this describes a permutation. In the general case of arbitrary m-n mappings there are recurring target position in the mapping set of different source positions and/or multiple target positions occurring in the mapping set(s) of some source positions. The set-permutation labels can be clustered into coarser categories of mapping complexity. We distinguish the following five cases, ordered by increasing complexity:

1. *Atomic*: If the alignment does not allow the existence of smaller (child) phrase pairs : a subset of alignment positions that is not connected to the other positions while also forming a contiguous sequence on the source and target does not exists.
2. *Monotonic*: If the alignment can be split into two monotonically ordered parts.
3. *Inverted*: If the alignment can be split into two inverted parts.
4. *PET (Permutation Tree)*: If the alignment can be factored as a permutation of more than 2 parts.
5. *HAT (Hierarchical Alignment Tree)*: If the alignment cannot be factored as a permutation of parts, but the phrase does contain at least one smaller phrase pair.

Typically there are multiple HATs for a word alignment, corresponding to different possible minimally branching factorizations of monotone parts. These alternative HATs can be efficiently computed and stored as a chart using a CYK-parser like chart parsing algorithm that parses the alignment and builds a hypergraph of HATs in the process.

A categorization of the complexity of the HAT as a whole is determined based on the complexity categories of the alignment mappings at its nodes. *Binary Inversion-Transduction Trees (BITTs)* is the least complex class consisting of only binary HATs that can be built for binarizable permutations [Huang et al., 2009], any HAT that contains only Monotonic and/or Inverted nodes belongs to this class. If a HAT contains at least one PET node but no HAT nodes it belongs to the category called PETs corresponding to general permutations [Zhang et al., 2008]. Finally the occurrence of at least one HAT node implies the set *HATs* which captures all possible many-to-many mappings.

[1] Note that while discontiguous translation equivalents exist, we limit us here to translation equivalents that are contiguous on both side (i.e. phrase pairs).

[2] Set-permutation labels for atomic nodes are omitted in the figures for reasons of readability

[3] Un-aligned words add no further constraints to the mapping, and thus can be ignored in the recursive composition of HATs.

Analysis/Results

Kind of HATs (S-permutations)	English-Dutch	English-French	English-German
BITTs (Binarizable permutations)	45.52%	52.84%	45.60%
PETs (Permutations)	52.63%	56.56%	52.55%
HATs (S-permutations)	100.00%	100.00%	100.00%

Table 1: The ratio of the different subsets of HATs in the corpus: BITTs, PETs and HATs

One way to characterize the complexity of hierarchical translation equivalence is to look what types of composition operators and associated types of Hierarchical Alignment Trees (HATs) aligned sentence pairs induce. The ratio of different subsets of HATs captures a coarse notion of the level of alignment complexity of real data. The table below shows this ratio for three aligned language pairs from Europarl, with approximately 1 million sentence pairs per language pair. Word alignment is done using GIZA++. This gives a taste of results obtained in our other studies. Much more extensive results are reported in [Sima'an and Maillette de Buy Wenniger, 2013, Maillette de Buy Wenniger and Sima'an, 2013].

References

References

[Huang et al., 2009] Huang, L., Zhang, H., Gildea, D., and Knight, K. (2009). Binarization of synchronous context-free grammars. *Computational Linguistics*, 35(4):559–595.

[Maillette de Buy Wenniger and Sima'an, 2013] Maillette de Buy Wenniger, G. and Sima'an, K. (2013). A formal characterization of parsing word alignments by synchronous grammars with empirical evidence to the itg hypothesis. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 58–67.

[Sima'an and Maillette de Buy Wenniger, 2013] Sima'an, K. and Maillette de Buy Wenniger, G. (2013). Hierarchical alignment trees: A recursive factorization of reordering in word alignments with empirical results. Internal Report.

[Zhang et al., 2008] Zhang, H., Gildea, D., and Chiang, D. (2008). Extracting synchronous grammar rules from word-level alignments in linear time. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 1081–1088.

Source Code

The source code and compiled executables are available at:

`https://bitbucket.org/teamwildtreechase/hatparsing`