

Rapid Adaptive Development of Semantic Analysis Grammars

Alicia Tribble, Alon Lavie, Lori Levin
Language Technology Institute, Carnegie Mellon University
5000 Forbes Avenue,
Pittsburgh, PA, USA
{atribble, alavie, lsl}@cs.cmu.edu

In this paper we describe a process for rapid development of semantic analysis grammars for Interlingual Machine Translation. The technique applies to existing systems and can be used to extend coverage into new languages quickly by separating the informant tasks performed by native speakers from the grammar writing tasks performed by engineers familiar with the system. A tool for automatic manipulation of parse trees uses information provided by both informant and engineer to create an example base of parse trees in the new language from which a grammar for that language can be read. Experimental results from a small-scale application of these tools are given to assess feasibility of the technique.

1 Introduction

Interlingual Machine Translation is an efficient strategy in applications where many different source and target languages will be paired, greatly reducing the number of transfer rules that are required. But even in this reduced situation, the creation of an analysis and generation grammar for each language is a time-consuming task that makes up a large portion of the engineering cost of the system.

Particularly at the stage when an infrastructure has been built and new languages are being added, the cost of grammar writing is high and can hold up development. One of the main reasons for this lies in the combination of skills required for the task. At such a point in the development of an MT system, the interlingua has evolved through use into a (reasonably) stable and complete representation of the language-independent concepts present in the domain at hand. The original grammar developers who design and test the interlingua are familiar with it to a degree that is difficult to communicate to new grammar writers. And yet native speakers of new languages must be recruited and trained in order to expand the system. The Janus MT system (Waibel 1996, Levin et al. 2000) is an example of a mature interlingual MT system in this position.

In this paper we describe a solution to this problem based on two principles: separating the tasks of grammar writing experts from those of native speaker informants, and increasing the rate of development with automatic tools for grammar induction. In this approach, the native speaker provides language-specific information in the form of translations, but he is shielded from the details of grammar writing. The grammar writer creates example structures that conform to the semantic representation requirements of the project but which serve as parse skeletons and have no language-specific

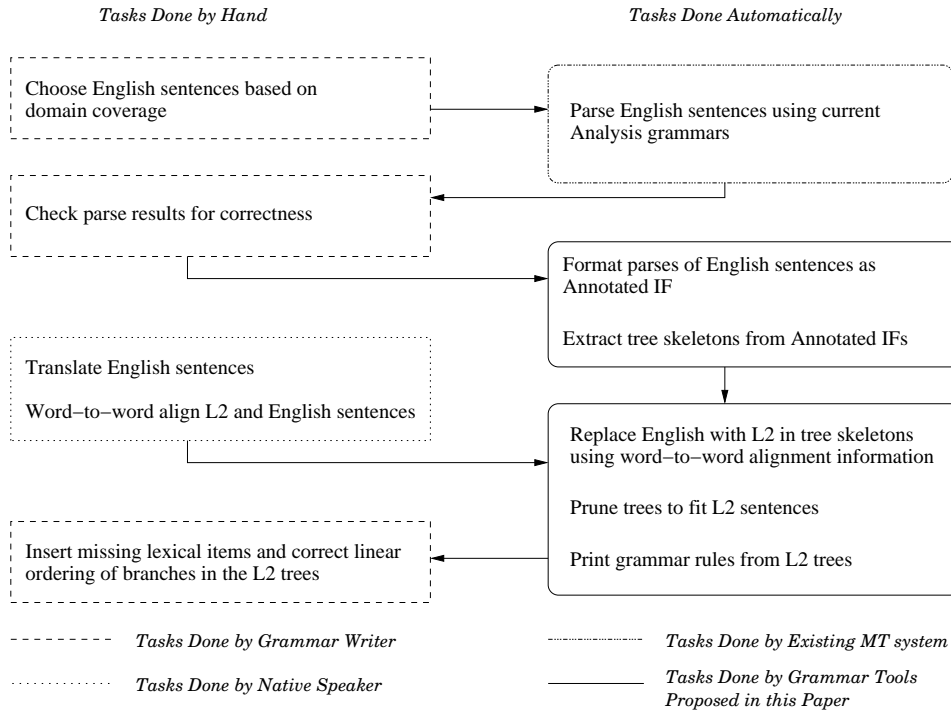


Figure 1: Grammar Adaptation Process

information. Finally, grammar induction tools link these two products together to form a forest of semantic parse trees and a grammar in the new language can be read from it. In the context of induction tools at large, our approach lies closer to the interactive techniques of Gavalda (2000) than to unsupervised approaches such as de Marcken’s (1995) and Lee’s (1996).

In Section 2 we discuss the steps of this grammar development process in detail. Section 3 describes the tools that are used to induce grammar rules from translation examples and the Machine Translation system that we use in our experiments. Although our research into this topic is continuing, we have used these tools to develop a small grammar for Polish and we describe the experimental results of this feasibility study in Section 4. Section 5 concludes the paper.

2 Approach

The grammar development process can be cleanly divided into tasks completed by a grammar expert, tasks completed by a native speaker of the new language (L2), and tasks completed automatically by grammar induction tools. Figure 1 shows the sequence of steps to developing a new grammar and who performs them.

2.1 Tasks for the Grammar Writer

This process starts with a delineation of the domain of the sentences that the new grammar will cover. In the context where our technique is applicable, grammars in

Table 1: Data labelling styles used in MT and Grammar Induction

English	feeling nauseous
Interlingual Form	c:give-information+concept (concept-spec=(body-state-spec=(feeling, nausea)))
Annotated IF	c:give-information+concept(concept-spec=(body-state-spec=(feeling(feeling))(nausea(nauseous))))

other languages already exist for this domain. An experienced developer of one of these grammars can provide a list of sentences representing the variety of concepts that are covered there.

The existing domain for our experiments in particular is medical dialogue between doctors and patients, and the interlingua is a semantic representation of the dialogue tasks achieved by a speaker in a given utterance. In our case, therefore, the body of sentences should include an example of every dialogue task. Our interlingua also uses semantic feature-value pairs to represent the detailed information in a sentence and each of these features should be represented in the example base as well.

One constraint on this process is the choice of L1 languages; since the L1 sentences must be translated by a native speaker of L2, choosing an L1 that is widely known increases the chance of finding a bilingual speaker to perform the translation. In our experiments the native informant is bilingual in Polish and English, and an existing English grammar was used for adaptation.

After an example base has been created in L1, it can be parsed using the existing system to yield an interlingua representation for each of the sentences. This representation is then reviewed for correctness by the grammar writer and annotated automatically to reflect the full tree structure of the semantic parse. In our experiments, analyzer output is already bracketed to show this tree structure, but because the output is interlingua the L1 surface strings have been removed. We therefore perform an automatic annotation of the interlingual format with L1 surface strings added to the frontier of the parse tree. We refer to the new format as Annotated IF; an example of this format is shown in Table 1.

2.2 Tasks for the Native Speaker

The responsibility of the native speaker in this procedure is limited to a detailed translation task. Since one of our goals is to spare ourselves the overhead cost associated with turning a native speaker into a grammar writer, we present him with the example L1 sentences without referring to their interlingual structure.

We instruct the native speaker to provide as many translations as possible for every English sentence. Although the amount of variation in the translated example base will vary from translator to translator, there is no reason to assume that this variation will be wider than the variation among hand-written grammars.

Additionally, the native speaker is responsible for providing word-level alignment information for every translated sentence. This is currently done in a second pass by editing a text file where the L1 and L2 sentences are given. The native speaker fills in an L2 alignment for each of the L1 words. Two examples from this file are given below.

34.1	feeling nauseous	39.2	i have had some warts on my left hand
34.1p	zle sie czuje	39.2p	miewalam brodawki na lewej rece
feeling	sie czuje	i	miewalam
nauseous	zle	have	miewalam
		had	miewalam
		some	
		warts	brodawki
		on	na
		my	
		left	lewej
		hand	rece

Example 34.1 contains a 1-N alignment in addition to word reordering. Example 39.2 contains an M-1 alignment and several 1-0 alignments but no word reordering. 0-1 alignments are also possible, as are M-1 alignments where the M words from L1 are not contiguous. In the second case, the native speaker is instructed to create M distinct sentence alignments for the sentence pair. In each of these sentence alignments one of the M words from L1 is shown as a 1-1 alignment with the word from L2; the others are left unaligned. In this way a separate link is established between each L1 word and the L2 target.

2.3 Tasks for the Automatic Grammar Development Tools

Given the set of parsed example sentences and their translations and word-level alignments, the grammar development process passes into a fully automatic stage where a tool written in C++ combines these inputs and produces a grammar file. The tool performs three basic operations. It replaces L1 surface strings in the example parse trees with their L2 translations, it prunes the resulting trees to fit the L2 sentences, and it prints grammar rules by reading them off of the trees. Finally, because these rules are read from a tree that was word-ordered according to L1, a reordering step must take place before they are usable for parsing new L2 sentences. This step can be done automatically or by the grammar writer. These steps are discussed in detail in Section 3.1.

The product of this automatic process is a file that can be used with the existing MT system to parse sentences of the new language.

3 Grammar Development Tools and the Existing MT System

3.1 Tree Manipulation Tools

We have developed a tool for performing the automatic steps described above that stores two types of data: an L1-L2 alignment table constructed from the text-based alignment file, and trees representing L1 parse structures.

3.1.1 Pruning

Tree transformations start with a search-and-replace of all L1 strings with their L2 equivalents. We examine every leaf node of the tree, looking up the L1 surface string in the alignment table. If the L1 string has a translation, we replace the L1 tokens with the L2 tokens and move on. If there is no L2 translation in the table, we delete the current leaf node.

At this point in the algorithm we have dangling internal nodes which should not be exposed on the frontier of the tree; these correspond to nonterminal grammar rules and not to surface strings. In order to maintain the correctness of the tree, we continue to traverse it in postorder and delete all childless nodes. In this way, nonterminals that are realized as surface strings in L2 are kept while nonterminals without surface representation are removed from the tree. A graphical representation of this tree pruning procedure is given in Figures 2 and 3.

3.1.2 Re-Ordering

After pruning, the series of L2 strings on the frontier of the IF tree may no longer form a legal utterance in that language. The L2 strings still appear in the order dictated by their English equivalents, which may not reflect their ordering in the original L2 translation. The L2 utterance may also be incomplete, since any L2 word that was not aligned to English has not yet been inserted.

We attempt to reconstruct the word order of the original L2 sentences by having our grammar writer adjust the rules manually using the translations as a guide. However, the operations she performs during this stage are limited to reordering the RHSs of individual grammar rules one at a time — she does not re-nest words or nonterminals under new LHSs or perform any other nonlinear operations. These restrictions on reordering make the task easy for a grammar writer to perform without any knowledge of L2, but more importantly they represent transformations on the rules that can be achieved through branch-switching operations on the parse trees, operations which can be implemented in software in a straightforward way.

In such an automated process, every leaf in the pruned tree is marked with its position in the original L2 example string, and out-of-order nodes are swapped at their lowest common ancestor until the frontier matches the L2 string in left-to-right ordering.

Finally, missing tokens from the L2 string must be inserted. These strings did not align to any of the L1 words and therefore have no well-defined position in the hierarchy of the tree. An unaligned L2 token can be inserted directly into the tree at the highest common ancestor of the tokens that appear to the left and right of it in the

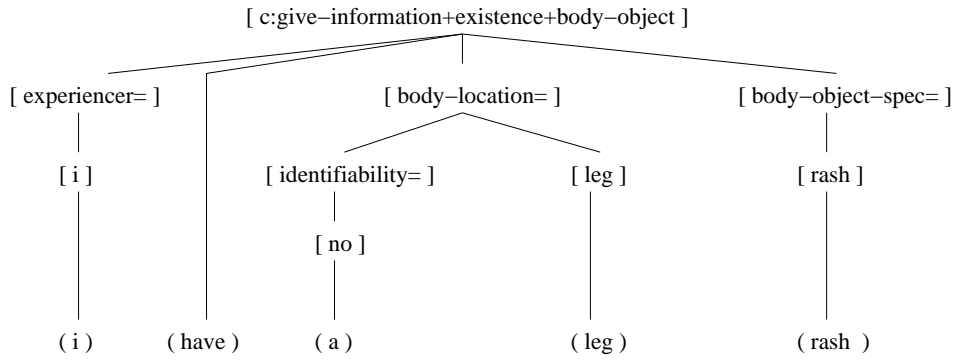


Figure 2: Annotated IF tree

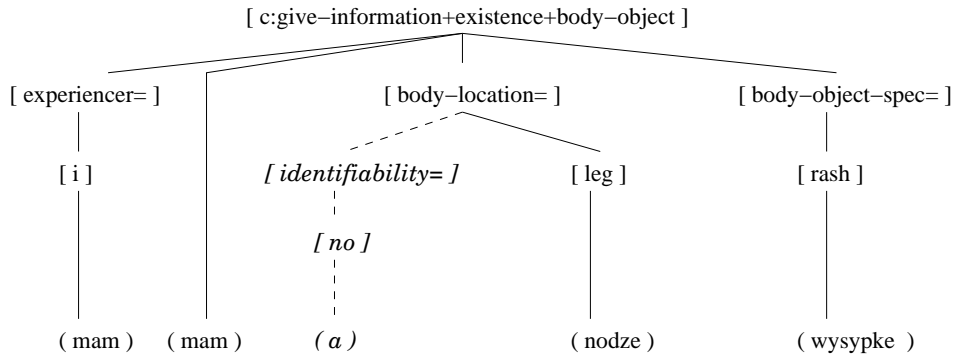


Figure 3: Tree after deletion

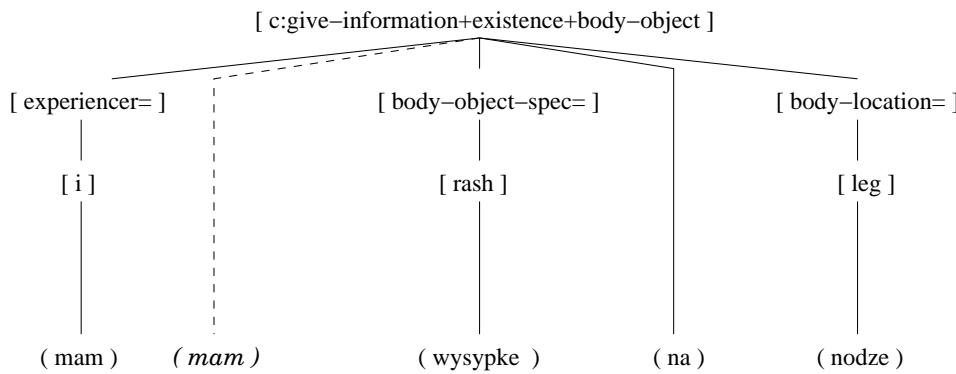


Figure 4: Tree after reordering

original L2 sentence. During this process the number of words on the frontier is also checked against the original sentence and repetitions that result from 1-N alignments are removed. Figure 4 shows the tree after branch-switching operations, demonstrating a successful recovery of the original L2 word order.

3.1.3 Rule Gathering

Once the IF trees have been aligned with the L2 strings, grammar rules can be read directly from the example trees. Every non-leaf node of the tree represents the LHS of a rule, with its children on the RHS. The technique of reading grammars from parse trees has been tested experimentally on syntactic treebanks with positive results (Charniak, 1996). The main difficulty with this approach for syntactic grammars has been the explosion in the number of resulting grammar rules, many of which are redundant. This is due to the fact that a single tree will generate many overlapping rules that may have been seen in previous examples. In our experiments we attempt to remove some of this redundancy by sorting the rules and merging them to produce a more concise grammar. Rules with identical LHSs are merged and the RHS of the new rule is the set of all unique patterns in the union of the original RHSs.

3.2 The Nespole! Project and the Interchange Format

The Machine Translation system for which we are developing grammars is a speech-to-speech system called LingWear. It includes a robust parser that works with context-free grammars. The interlingua is based on the Interchange Format (IF), a representation developed in another speech translation project called Nespole!.

The Nespole! IF is a task-based representation of the semantics of a unit of speech. Since the system translates spoken dialogue, these units are called Spoken Dialogue Units (SDUs), and they range in length from a single word (“hello”) up to a full sentence (“I’d like a room”). The interlingual tag for a single SDU is composed of four parts: a Speaker Label, a Speech Act, a Concept List, and a list of Arguments and values.

<u>c:</u>	<u>give-information</u>	<u>+plan+trip</u>	<u>(who=i,destination=italy,time=(month=12))</u>
Speaker	Speech Act	Concept List	Argument List
Tag			

The arguments list contains the most detailed information in the IF representation and its form is strictly hierarchical. Generally, we can interpret any IF label as a tree structure, with the Domain Action at the root. Subsequent nodes represent arguments, sub-arguments, and values whose children are sub-arguments, values, or bottom-level token nodes, respectively. This tree-structured property, along with the maturity and thorough specification of the IF, make it an excellent candidate for our experiments.

4 Experimental Results

As a preliminary experiment on the feasibility of this approach, we created a new grammar for the LingWear system in Polish using the tools described above. Figure 1,

which we have used to explain the steps of our grammar induction process, models the procedural details of this experiment.

Our goal was to cover a subset of the LingWear domain including 11 Domain Actions that refer to the existence of physical symptoms. An example base of 41 English sentences with a vocabulary size of 95 tokens was chosen to represent this subdomain. A native speaker of Polish translated and aligned 77 Polish sentences to the English example base with a resulting vocabulary size of 115 tokens.

We returned to the native speaker several weeks later and asked him to generate a new set of sentences in the same domain, using the vocabulary list as a guide but attempting not to repeat constructions from the training set. The result was a set of 39 new Polish sentences which were used for evaluation.

The English parse trees and English-Polish alignments were processed automatically by the tree adaptation routines described above, and the resulting grammar rules were reordered by hand by the grammar writer using the translations as a guide.

The resulting Polish grammar contained rules for each of the 11 Domain Actions, plus 60 nonterminal rules for arguments and values. Two example rules from this grammar are given below.

```
s[request-information+existence+body-object]
  (czy [body-object-spec=])
  (czy jest [body-object-spec=])
  (czy ma pan [body-object-spec=] [body-location=])
```

```
[body-object-spec=]
  ([symptom-blood])
  ([wart@quantity=plural])
  ([cramp@quantity=plural])
  ([rash])
  ([gas])
  ([bile])
  ([ulcer])
  ([whose=i] [arm])
  ([whose=i] [side=] [body-foot])
```

Parsing the Polish test set with this grammar resulted in a parse for 33 of the 39 sentences. Our robust parser produces some output even when parts of the input string must be skipped, so some of these parses represent coverage of only a portion of the test sentence. The average coverage over all of the input sentences was 51.7%, with 6 sentences covered perfectly (100%) and 6 not covered at all (0%).

The real concern in analysis grammar evaluation is whether an accurate translation can be generated from the analysis result. Close inspection of the parse output reveals that many of the partial parses contained enough information for an accurate translation, indicating that 52% coverage is a lower-bound estimate of the translation quality. An example of such a parse is shown here: notice that although only 25% of the surface

string was parsed, it was labelled with the correct Domain Action and would produce an accurate translation (“It is on my leg”).

```
; Parsing utt 5 (line 5)
; "<s> to jest na nodze </s> "
; Interpretation 5.1
; !<s> !to !jest !na nodze !</s>
; Coverage 25% (1/4) in 1 tree
[give-information+body-object]::MED
  ( [body-location=]::MED ( [leg]::MED ( nodze ) ) )
```

5 Conclusions

In conclusion, we found the results of our preliminary study promising for the rapid development of grammars in new languages. The overall time to development for our test grammar was on the order of 1-2 days. This includes native speaker time and grammar writer time, and the process we describe here allowed the tasks to be completed independently by the two experts without additional cost for coordinating their schedules. We also note that much of the grammar writer’s effort was spent selecting and formatting the example base of English sentences; this example base is an artifact that can be reused in all subsequent experiments in this domain, further reducing development time for grammars in other new languages.

In the future we plan to apply this technique in coordination with machine learning algorithms for grammar expansion, in an effort to cover larger domains with little additional human effort. One candidate for such expansion is a grammar generalization tool developed by Ben Han for Nespole! grammars in (Lavie et al., 2001).

This grammar development process adds to the growing number of systems for rapid-deployment Machine Translation and contributes positively to an increasingly important field.

References

- Eugene Charniak: 1996, ‘Tree-bank Grammars’, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence, 1031–1036*, Menlo Park 1996.
- M. Gavalda: 2000, ‘Epiphenomenal Grammar Acquisition with CSG’, in *Proceedings of the Workshop on Conversational Systems of the 6th Conference on Applied Natural Language Processing and the 1st Conference of the North American Chapter of the Association for Computational Linguistics (ANLP/NAACL–2000)*, Seattle, U.S.A., 2000.
- A. Lavie, L. Levin, T. Schultz, C. Langley, B. Han, A. Tribble, D. Gates, D. Wallace, and K. Peterson: 2001, ‘Domain Portability in Speech-to-speech Translation’, in *Proceedings of the First International Conference on Human Language Technology Research (HLT–2001)*, San Diego, CA, March 2001.
- Lillian Lee: 1996, ‘Learning of Context-Free Languages: Survey of the Literature’, in *Technical Report TR–12–96, Center for Research in Computing Technology, Harvard University*, Cambridge, MA, 1996.

de Marcken:1995 Carl de Marcken: 1995, 'Lexical Heads, Phrase Structure, and the Induction of Grammar', in *Third Workshop on Very Large Corpora, 14-26*, Cambridge, MA, 1996.

Waibel, Alex, Michael Finke, Donna Gates, Marsal Gavalda, Thomas Kemp, Alon Lavie, Lori Levin, Martin Maier, Laura Mayfield, Arthur McNair, Ivica Rogina, Kaori Shima, Tilo Sloboda, Monika Woszczyna, Torsten Zeppenfeld, and Puming Zhan: 1996, 'JANUS-II: Translation of Spontaneous Conversational Speech', in *Proceedings of ICASSP-1996*