Translingual Grammar Induction

John Lee and Stephanie Seneff

Spoken Language Systems
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139 USA
{jsylee,seneff}@sls.csail.mit.edu

Abstract

We propose an induction algorithm to semi-automate grammar authoring in an interlingua-based machine translation framework. This algorithm uses a pre-existing one-way translation system from some other language to the target language as prior information to infer a grammar for the target language. We demonstrate the system's effectiveness by automatically inducing a Chinese grammar for a weather domain from its English counterpart, and showing that it can produce high-quality translation from Chinese back to English.

1. Introduction

For more than a decade, our group has been conducting research leading to the development of multilingual conversational systems. These systems enable naive users to access and manage information using spoken dialogue in a variety of languages. In the language understanding component, a common meaning representation, or *interlingua*, is extracted from the user input. This language-independent representation facilitates effective communication with the application back-end, the dialogue management and the discourse context resolution components.

Within this framework, we have recently introduced a language learning system [1]. A native speaker of Chinese who wishes to learn English, for example, can speak a sentence in his/her native tongue and have the system paraphrase it in English. He/she can then attempt to repeat the English sentence to advance a dialogue with the system in English. Two questions arise from this research: (1) is our interlingua-based framework effective for translation, at least in restricted domains? and if so, (2) are there ways to quickly develop grammars that extract a meaning representation from user input in multiple languages? Currently, grammar authoring is a laborious, error-prone process that demands a lot of expertise and patience.

In many domains of interest to us, we already have mature, high-quality grammars in place for English. In this paper, we propose a grammar induction algorithm that leverages these grammars to semi-automate grammar authoring in other languages. We then describe experiments demonstrating that an induced grammar can generate high-quality translation in a restricted domain.

2. System Description

Our conversational system takes two parameters for each language: an *understanding grammar* (PARSE_L) for our natural language understanding system, TINA [2], which maps a sentence in language L to a common meaning representation; and a *generation module* (GEN_L), which verbalizes a meaning representation in language L. Thus, to add a new language L' to the system, one needs to implement both $\text{PARSE}_{L'}$ and $\text{GEN}_{L'}$.

Like most NLU systems, TINA uses a set of context-free rules to describe the sentence structure. The grammars that are designed for our multilingual conversational systems typically incorporate both syntactic and semantic information simultaneously. At the higher levels of the parse tree, major syntactic constituents, such as subject, predicate, object, etc., are explicitly represented through syntax-oriented grammar rules. The syntactic structures tend to be domain-independent, capturing general syntactic constraints of the language. Near the leaves of the parse tree, major semantic classes, such as *weather_verb*, *date_name*, etc., are constructed according to semantic-oriented grammar rules. The semantic structures tend to be domain-dependent, capturing specific meaning interpretations in a particular application domain. Such a grammar is able to combine syntactic and semantic constraints seamlessly. It also offers an additional convenience that no separate semantic rules are necessary for meaning analysis. The semantic representation can be derived directly from the resulting parse tree. Fig 1 shows an example of a parse tree.

<table>
<tr><td colspan="4">question</td></tr>
<tr><td colspan="4">do_question</td></tr>
<tr><td rowspan="2">will</td><td rowspan="2">subject
it</td><td colspan="2">predicate
predicate_v
intr_vp</td></tr>
<tr><td>intr_vb
weather_vb</td><td>vb_args
when
date_name</td></tr>
<tr><td>will</td><td>it</td><td>rain</td><td>tomorrow</td></tr>
</table>

Figure 1: *Parse tree for 'Will it rain tomorrow?'*

The parse tree serves as a stepping stone towards the meaning representation, which, in our system, is a *semantic frame*: a hierarchical structured object that encodes meaning. Designated nodes in the tree guide a process to create frames or assign key values in the frame under construction. For example, the *do_question* node creates a **verify** frame, and the *will* node assigns its leaf as the value for the *auxil* key. Fig. 2 shows the semantic frame produced by the tree in Fig. 1.

The generation module [3] maps a semantic frame to a surface string. It specifies the order in which components in the frame are to be processed into substrings, and consults a gener-

This work is in part supported by a fellowship from the National Sciences and Engineering Research Council of Canada, and by the NTT Corporation.

```
{ verify
  auxil "will"
  topic { pronoun
          name "it" }
  pred { rain
         pred { temporal
                topic { weekday
                        name "tomorrow" } } } }
```

Figure 2: *Semantic frame for 'Will it rain tomorrow?'*

ation lexicon to obtain surface-form mappings. Fig. 3 shows the steps taken in a simple Chinese generation module to generate the string "ming2 tian1 hui4 xia4 yu3 ma5" from the semantic frame in Fig. 2.

3. Approach

We propose an induction algorithm that automatically infers $\text{PARSE}_{L'}$, given the following three pieces of prior information:

- TRAIN_L: Training sentences in some other language L.
- PARSE_L: This understanding grammar is used to parse the sentences in TRAIN_L. The L parse trees are then transformed by a series of operations (see §5) into L' parse trees. Context-free rules read off the resulting treebank constitute $\text{PARSE}_{L'}$.
- $\text{GEN}_{L'}$: This generation module paraphrases the semantic frames produced by PARSE_L into the L' language, and simultaneously infers an L-L' word alignment (see §4). It also sheds light on the structure of the L' language, which is crucial in the tree transformation steps.

In theory, the development effort needed for adding a new language L' to the conversational system is then reduced to $\text{GEN}_{L'}$.

In the rest of the paper we illustrate this induction process with an example where L is English and L' is Chinese.

4. Word Alignment

The semantic frame serves as the link between the L and L' sentences. During parsing, we align L words to components in the frame; during generation, we align components in the frame to L' words.

When the tree in Fig. 1 produces the semantic frame in Fig. 2, we remember the nodes that are responsible for creating each component. For example, the word "will" is aligned to the key *auxil*. When the semantic frame is verbalized to an L' string, we could similarly observe the L' words emitted from each component of the frame. For example, the *auxil* key is aligned to the word "hui4". This yields an L-L' word alignment, as shown in Fig. 4.

Generation step	Surface string
1. Verbalize any **temporal** predicates	ming2 tian1
2. Verbalize the *topic* frame, except if the predicate is **rain**	*null*
3. Verbalize any *auxil* key	hui4
4. Verbalize predicates	xia4 yu3
5. Add *question particle* if the main clause is **verify**	ma5

Figure 3: *Generation steps for the Chinese paraphrase, "ming2 tian1 hui4 xia4 yu3 ma5"*

```
{ verify [L=null, L'=ma5]
  auxil [L=will, L'=hui4]
  topic { pronoun
          name [L=it, L'=null] }
  pred { rain [L=rain, L'=xia4 yu3]
         pred { temporal
                topic { weekday
                        name [ L=tomorrow,
                               L'=ming2 tian1] } } } }
```

Figure 4: *L-L' word alignment for 'Will it rain tomorrow?'*

5. Tree Transformation

5.1. Leaf Translation

The first step in the transformation from an L parse tree to an L' parse tree is to translate the leaves based on the word alignment obtained in §4. If an L word is aligned to one or more L' words, then we simply overwrite its leaf with the L' translation. For example, we replace "tomorrow" with "ming2 tian1".

5.2. Branch Pruning

If an L word is not aligned to any L' word, we prune its branch. Hence some information may be lost. For example, the word "it" has no equivalent in the Chinese paraphrase. After removing its branch from the parse tree, the topic **pronoun** in the semantic frame would also be lost.

5.3. Branch Movement

Next, the branches are re-ordered to match the L' word order. A simple-minded approach would lead to the strange-looking parse tree in Fig. 5. This tree would suggest, for example, that "ming2 tian1" could by itself be parsed under *intr_vp*. Furthermore, the semantic frame produced by this tree would have a different hierarchical structure than the one in Fig. 2; namely, the **temporal** predicate would be placed at the top-level frame rather than under the **rain** predicate.

We make use of TINA's trace mechanism [2] to tackle this problem. From our point of view, a trace is necessary when the word orders of L and L' are so different that it is impossible to go from one to the other without changing the hierarchical structure of the parse tree. By marking the "tomorrow" branch as *extraposed*, we indicate that it is to be detached and grafted to the *extrapose* node in the "*trace*" column after parsing. The final parse tree is shown in Fig. 6.

5.4. Branch Insertion

Finally, L' words that are not aligned to any L words are inserted. The new branch may be attached to its left or right neighbor, or to the lowest common ancestor of the two neighbors. For instance, "ma5" could be attached under *intr_vp* as the right sibling to *vb_args*, as well as under *do_question* as the right sibling to *predicate*. We turn to its generation history to make the decision. Since "ma5" is generated by **verify**, the top-level frame, we infer that it is not dependent on "xia4 yu3" (rain), but on the

question			
do_question			
predicate predicate_v intr_vp vb_args when date_name	will	predicate predicate_v intr_vp intr_vb weather_vb	question particle
ming2 tian1 (tomorrow)	hui4 (will)	xia4 yu3 (rain)	ma5 (*null*)

Figure 5: *Parse tree without trace*

question				
do_question				
extraposed date_name	will	predicate predicate_v intr_vp		question particle
		intr_vb weather_vb	vb_args when **extrapose**	
ming2 tian1 (tomorrow)	hui4 (will)	xia4 yu3 (rain)	*trace*	ma5 (*null*)

Figure 6: *Parse tree with trace included*

whole sentence. It is thus attached under *do_question*, with the label *question particle* taken from the generation module.

6. Experiments

We tested this induction approach on the JUPITER weather information domain [4], our most mature domain to date, using English as L and Chinese as L'.

6.1. Data Preparation

We have a large corpus of English utterances of naive users asking about weather over the phone. Since it is critical for the induction algorithm to learn from valid English parse trees and Chinese paraphrases, we filtered this corpus in three stages. First, we filtered out sentences to which our English grammar, PARSE_e, could not give a complete parse. Next, we obtained Chinese paraphrases for the remaining sentences using our Chinese generation module, GEN_c. Since this generation module was not yet entirely mature, we further tested the quality of these paraphrases. We filtered out those paraphrases that could not be parsed by a previously authored Chinese grammar[1]. Finally, we manually filtered out those paraphrases that translated only a part of the original English sentence.

At the end of the process, we were left with a little over 7000 sentences. We randomly set aside roughly 10% of these sentences for testing.

6.2. Evaluation Metric

Given a pair of sentences in English and Chinese, we used PARSE_e and the induced PARSE_c, respectively, to produce two semantic frames, say f_e and f_c. We then obtained their English paraphrases, $\text{GEN}_e(f_e)$ and $\text{GEN}_e(f_c)$. Next, we calculated the word error rate of $\text{GEN}_e(f_c)$ when compared with $\text{GEN}_e(f_e)$, which we considered as the "gold standard". If PARSE_c failed to parse the sentence, $\text{GEN}_e(f_c)$ would be *null* and hence given a 100% deletion error. The average length of $\text{GEN}_e(f_e)$ in the test set is 6.0 words.

[1] In general, such a grammar of course would not pre-exist, and some other methods, such as manual assessment, would be required here.

Error	Word	%	Error	Word	%
del	is	9.6%	sub	any → a	1.7%
del	the	7.9%	sub	is → does	1.7%
sub	for → in	7.0%	del	today	1.6%
del	be	6.2%	sub	will → is	1.5%
del	for	4.8%	ins	tomorrow	1.5%
del	it	4.8%	del	in	1.4%
del	like	3.9%	sub	how → what	1.4%
ins	about	2.4%	del	tomorrow	1.3%
ins	today	2.0%	ins	now	1.3%
del	will	1.9%	sub	are → is	1.1%

Table 1: *Twenty most frequent errors in the English paraphrases, as percentages of the total error (del = deletion, sub = substitution, ins = insertion)*

6.3. Results

Fig. 7 shows the learning curve of the induction algorithm. The best induced grammar performed at 27.3% word error rate (15.5% deletion, 7.7% substitution and 4.1% insertion rate). Table 1 lists the 20 most frequent errors, which collectively accounted for 65.0% of the total error.

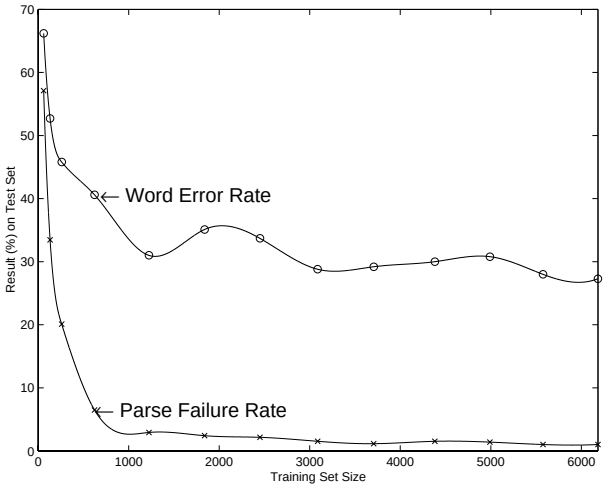

Figure 7: *Performance of induced grammar with respect to size of training data*

The word error rate is a rather harsh measure for translation quality. In many cases different phrase orderings resulted in high error rates in paraphrases that were entirely acceptable. For example, the following pair of $\text{GEN}_e(f_e)$ and $\text{GEN}_e(f_c)$ incurred a 43% error rate.

> What is *the* temperature in England *tomorrow*?
> What is temperature *tomorrow* in England?

Almost all deletions of "tomorrow" or "today" in $\text{GEN}_e(f_c)$ were coupled with insertions of the same words elsewhere in the paraphrase.

Aside from "tomorrow" and "today", none of the other words in Table 1 are content words that significantly alter the meaning of the paraphrase in the weather domain. The majority of the errors were deletions of words that were in fact absent from the Chinese paraphrases. The induction algorithm therefore pruned the branches of these words, leading the induced PARSE$_c$ to produce impoverished semantic frames. Such deletions would exist even for a grammar developed by an expert. It should be the responsibility of GEN$_e$ to reinstate such missing features based on first principles and/or statistical methods. We are developing a generation preprocessor [5] for this purpose.

Other errors were caused by translation variants of Chinese words in the domain. In the experiment, we simply selected the variant that was seen most often in the alignments. For instance, "zhi1 dao4" translates to the more frequently occurring "know about" rather than to "know", accounting for most of the insertion errors for "about".

Finally, mistakes in word alignment introduced some noise and redundancies to the grammar.

7. Related Work

Grammar induction can be defined as the process of inferring the structure of a language L', given a corpus of sentences drawn from it. In nearly all cases, some prior information, such as existing grammars in related domains or languages, is often used as a starting point.

In [6] and [7], the prior information consists of a set of fundamental concepts (e.g., time, date) that are useful in multiple domains. In [8], the prior information is a simple "domain model", which is progressively expanded and refined as the system elicits new examples from the user.

In [9], which is most closely related to our work, the prior information is a grammar for some language L. A native speaker of L' provides pairs of aligned sentences in L and L'. The induction algorithm transforms the L parse trees into L' parse trees. With no knowledge of the structure of the L' language beyond the word alignments, the algorithm is sometimes forced to make rather arbitrary assumptions, especially when reordering branches and inserting new ones. The algorithm was used to induce a Polish grammar from an English grammar in a domain for physical symptoms. On a test set of 39 sentences, the induced grammar achieved 52% coverage of key-value pairs in the meaning representation.

8. Future Plans

We plan to further our research in many directions, including:

1. We anticipate that a developer will need to make adjustments to an induced grammar. After the developer has improved the grammar, s/he may later want to extend its coverage to more sentences in the domain. We would like to enable the induction algorithm to carefully add new induced rules to the modified grammar, while respecting the changes made by the developer.
2. We are presently developing generation modules for Mandarin, French, Japanese, Spanish and Korean in the PHRASEBOOK domain, intended for tourists who do not speak the language in their destination countries. In the future we plan to expand to Arabic and Urdu. This domain will be incorporated into our language learning system. Both languages used in our experiment, English and Chinese, are subject-verb-object languages. We would like to see the performance of this induction approach when L and L' have very different word ordering, such as English and Japanese.
3. The current algorithm is very sensitive to correct L-L' word alignment. If GEN$_{L'}$ is not yet working well, the quality of the induced grammar degrades significantly. A statistical treatment on word alignment may be warranted.
4. While we have thus far only evaluated the induced grammar on paraphrases into *natural* languages, we are also interested in applying the grammar in spoken dialogue applications, where the system must understand the query and respond appropriately. We generally use a 'paraphrase' into a flattened (key: value) representation to transform the semantic frame into a format that is more transparent to the dialogue manager. Formal evaluation of the differences in this (key: value) representation could help us judge the effectiveness of our generated grammars for dialogue interaction.

9. References

[1] Seneff, S. "Spoken Conversational Interaction for Language Learning", To appear in Proceedings of the InSTIL Symposium, Venice, Italy, 2004.

[2] Seneff, S. "TINA: A Natural Language System for Spoken Language Applications", Computational Linguistics, 18(1):61–86, 1992.

[3] Baptist, L. and Seneff, S. "GENESIS-II: A Versatile System for Language Generation in Conversational System Applications", in Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP-2000), pages 271-274, Beijing, China.

[4] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T. and Hetherington, L. "JUPITER: A Telephone-Based Conversational Interface for Weather Information" IEEE Trans., Speech and Audio Proc., 8(1), pages 85-96, 2000.

[5] Cowan, B. "Tailoring Meaning Representations for Machine Translation using PLUTO" in Proceedings of the MIT Student Oxygen Workshop, Gloucester, MA, September 2003.

[6] Wang, Y.-Y. and Acero, A. "Combination of CFG and N-gram Modeling in Semantic Grammar Learning", in Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003), Geneva, Switzerland, September 2003

[7] Lavie, A., Levin, L., Schultz, T., Langley, C., Han, B., Tribble, A., Gates, D., Wallace, D. and Peterson, K. "Domain Portability in Speech-to-Speech Translation", in Proceedings of the Human Language Technology Conference (HLT-2001), San Diego, CA, March 2001.

[8] Gavaldà, M. and Waibel, A. "Growing Semantic Grammars", in Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL-98), Montréal, Canada, August 1998.

[9] Tribble, A., Lavie, A. and Levin, L. "Rapid Adaptive Development of Semantic Analysis Grammars", in Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-2002), Keihanna, Japan, March 2002.