

Learning Associations Between Grammars: a New Approach to Natural Language Understanding

Enrique Vidal[†] Roberto Pieraccini & Esther Levin

Speech Research Department, AT&T Bell Labs. 600 Mountain Avenue, Murray Hill, NJ 07974, USA

[†]On leave from Dep. Sistemas Informaticos y Computacion, Univ. Politecnica de Valencia, 46071 Valencia, Spain.

ABSTRACT

Language Understanding can be seen as a process of translation from input natural language sentences into commands of a certain Semantic Language that drive the actions associated to the meaning of the sentences. Under this point of view, a new approach is introduced to automatically learn the required transducers from training sets of input-output examples. This approach overcomes certain input-output sequentiality problems of previous techniques. Experiments are presented with a subset of the DARPA ATIS corpus that show the capabilities of the new method to learn useful English-semantic mappings for this task.

1. Introduction.

A completely general definition of a Speech Understanding system (SUS) is that of a machine accepting strings of words (or acoustic features) as input and producing sentences from a certain "Semantic Language" (SL) that specify the actions to be performed. In most of the SUS's we may envision to develop in the near future, SL will be just a convenient (computer, i.e., formal) command language used to actually drive the final actions required for the considered task. Under such a point of view Language Understanding is a process of (formal) Transduction. The problem then arises of how to obtain an adequate transducer with the required input-output behavior.

Let nl be a natural language (NL) sentence and sl a representation of its meaning expressed in an appropriate SL. Provided that a training set of pairs (nl, sl) is available, methods previously developed by the authors [8-10] can be used for learning the components of an understanding system under the semantic transduction framework. Although reasonably good results have been obtained with these methods, some drawbacks have also been identified. The most important can be summarized as follows: first, there must be a *sequential correspondence* between symbols (i.e. words) in an nl sentence and symbols in the related sl ; second, an initial *segmentation* of each nl in terms of the basic semantic units (*concepts*) that appear in its corresponding sl is required for initializing the training process.

Given a particular understanding task, the first of these requirements can be met, in principle, by appropriately designing an "ad hoc" SL for this task [8-10] However, this is not always easy and one should rather prefer to be able to

adopt a more general SL and/or one that is already available and has proved convenient for the task. Clearly, it may then happen that the sequential correspondence requirement is not satisfied. Also, the second requirement can be fairly impractical when a large database of examples is available, or it can be impossible in the case of non sequential SL.

Here, we propose a new method that does not require an initial segmentation nor any assumptions on the positional relationships of the symbols in the training pairs. This method was tested with an experiment in which a system was trained to translate sentences of the ATIS task [6] into the corresponding semantic transcription in terms of "pseudo-English" formal queries. The results show the capability of this method to automatically learn useful NL-SL mappings for this task.

2. Language Understanding and Formal Transduction: Sequentiality issues.

A block diagram of our view of Language Understanding as a transduction process is depicted in Fig.1. While the great simplicity and generality of this view may be conceptually very pleasant, the actual development of real systems under this paradigm is not easy. The main problem is how to learn or "train" the transducer T from a training set of input-output examples. In trying to simplify such a problem, the single-block diagram of Fig.1 can be split into a two-block diagram as shown in Fig.2. The first block is called *Semantic Transducer* (ST). It implements the actual understanding function by translating the input speech or text into a "meaning" represented in an *Intermediate Semantic Language* (ISL). The second block is a "*Representation Converter*" (RC) that deterministically converts the ISL representation of the meaning into a target formal command which would actually drive the corresponding action.

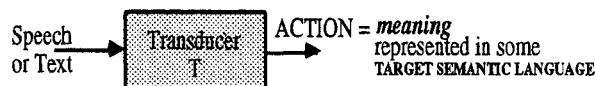


Fig. 1. Language Understanding: basic scheme.

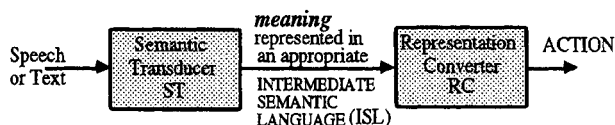


Fig. 2. Language Understanding: a simpler block diagram.

Clearly the freedom that entails the possibility of choosing an appropriate ISL can greatly simplify the learning of the Semantic Transducer ST. However this freedom is limited by three important design constraints:

1. ISL must be simple enough in order to make the learning of ST affordable.
2. ISL should strictly impose the actual semantic constraints of the given task.
3. ISL must be close enough to the target Semantic Representation (action driver) not to make exceedingly difficult the implementation of RC.

In trying to meet constraint #1, we have recently introduced the use of ISL's that are *sequential with the input*. This sequentiality allows us to perform a *Semantic Segmentation* of the input speech or text which makes well known techniques readily available for solving the ST learning problem. Using this idea, we have approached tasks such as learning to "understand" numbers, spontaneous English queries to the ATIS database [6] and spontaneous Spanish queries to the BDGEO database [4]. However, while perfectly adequate ISL's could be easily adopted for the rather restricted "number understanding" task [10-11], only partially appropriate ISL's were found for the ATIS task [8-9] or for (restricted versions of) the BDGEO task [11]. The main concern is that the sequentiality assumption often prevents the ISL to be expressive enough to correctly cover the underlying semantic space and/or to actually introduce the required semantic constraints.

It is interesting to notice that the sequentiality assumption can be largely relaxed by the use of (finite-state) "subsequential transducers" for which a learning algorithm has been recently introduced [7] and successfully applied to many artificial and pseudo-natural tasks [7] [3]. Also limited, though encouraging, success has been achieved by using this technique in natural tasks such as ATIS [9]. Nevertheless, as input-output sequential correspondences become weaker, finite-state transduction schemes become increasingly inappropriate; either because the amount of non-sequentialities lead to huge models, or simply because the exceedingly unsequential nature of the underlying mapping is beyond the capabilities of these models.

3. Beyond Sequentiality: The new approach

Transduction schemes more powerful than finite-state models have been studied in the Theory of Formal Languages [1], but the possibility of learning such devices from training data does not seem to have been explored so far. On the other hand, a more pragmatic approach has been proposed in [2], which entails a direct statistical modeling of the relations between input and output tokens and their relative ordering. Here we will introduce a new approach which does not assume or require any kind of explicit modeling of ordering or relations between input-output tokens in the corresponding input and output sentences. Instead, token order is assumed to be modeled by the corresponding Input and Output Language Models (stochastic grammars), while the relations between these tokens are *indirectly* taken into account by assuming *associations between the non-terminals or rules of the input and output grammars*.

Let G' , G be these input and output grammars, and $\mathcal{L}(G)$, $\mathcal{L}(G')$ their associated languages (NL,SL), respectively. We can look at our translation task as a reverse generation problem in which sequences of $\mathcal{L}(G)$ are somehow converted into sentences of $\mathcal{L}(G')$ that we can observe. Given one of these sentences $x \in \mathcal{L}(G')$, the problem is to find an $y \in \mathcal{L}(G)$ that most likely produced x ; i.e., one for which $P(y|x)$ is maximum. By applying Bayes' rule, this can be written as:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{L}(G)} P(x|y) P(y) \quad (1)$$

Here, $P(y)$ is directly supplied by the Output Language Model (i.e., $P(y)=P(y|G)$), while $P(x|y)$ is the actual translation probability that we aim to represent through "grammar association". To this end, let $D_{G'}(x)$, $D_G(y)$ be derivations of x and y through G' and G , respectively. If G' and G are *unambiguous* grammars, then $D_{G'}(x)$ and $D_G(y)$ are unique. Otherwise, we can just adopt a Viterbi-like approximation and choose unique *max-likelihood* derivations for $D_{G'}(x)$, and $D_G(y)$. Therefore, a one-to-one correspondence between strings and derivations is established and we can exchange x for $D_{G'}(x)$ and y for $D_G(y)$. Correspondingly our optimization problem (1) can be rewritten as:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{L}(G)} P(D_{G'}(x) | D_G(y)) P(y) \quad (2)$$

where $P(D_{G'}(x) | D_G(y))$ is the probability that derivation $D_{G'}(x)$ is produced, given that y is derived as $D_G(y)$. Unfortunately, this probability may entail complex probabilistic relations between individual rules and/or non-terminals of the input and output grammars and does not seem to admit a simple factorization which is also correct and convenient. Nevertheless, we have adopted the following crude heuristics which has proved useful in practice:

$$P(D_{G'}(x) | D_G(y)) \approx \prod_{r \in D_G(y)} P_D(r) \quad (3)$$

with

$$P_D(r) = \prod_{r' \in D_{G'}(x)} P(r'|r) \cdot \prod_{r' \notin D_{G'}(x)} P(\text{not } r'|r)$$

where $P(r'|r)$ and $P(\text{not } r'|r)$ are, respectively, the probability of using and *not* using r' in an input derivation given that r participates in an output derivation. These probabilities can be estimated from frequency counts of the corresponding events in the (max-likelihood) derivations of training pairs through the input and output grammars.

The motivation for this heuristics comes from the fact that deterministic conditions for rules of G to participate in the parsing of output sequences can often be written just as simple *conjunctive Boolean formulae* which involve only terms stating the use of rules of G' in the parsing of input strings and the corresponding negated terms (*not* use of input rules). Examples of these conditions can be observed in Fig.3, where Input (G') and Output (G) grammars are shown for the translation of a small set of English numbers into their Roman-numeral representation. For instance, the use of rule $4 \rightarrow 7$ ($r_{4,7}$) of the Roman-numeral grammar can be conditioned as follows:

$$\begin{aligned}
r_{4,7} &= \rho_{1,7} \wedge \rho_{7,9} \wedge \bar{\rho}_{9,16} \wedge \\
&\bar{\rho}_{1,2} \wedge \bar{\rho}_{2,16} \wedge \bar{\rho}_{1,3} \wedge \bar{\rho}_{3,16} \wedge \bar{\rho}_{1,4} \wedge \bar{\rho}_{4,16} \wedge \\
&\bar{\rho}_{1,5} \wedge \bar{\rho}_{5,9} \wedge \bar{\rho}_{1,6} \wedge \bar{\rho}_{6,9} \wedge \bar{\rho}_{1,8} \wedge \bar{\rho}_{8,9} \wedge \\
&\bar{\rho}_{6,10} \wedge \bar{\rho}_{7,10} \wedge \bar{\rho}_{8,10} \wedge \bar{\rho}_{10,16}
\end{aligned}$$

where $\rho_{i,j}$ ($\bar{\rho}_{i,j}$) is a predicate whose value is *true* (*false*) if rule $i \rightarrow j$ of the English grammar has been (has not been) used in the parsing of an input string and *false* (*true*) otherwise.

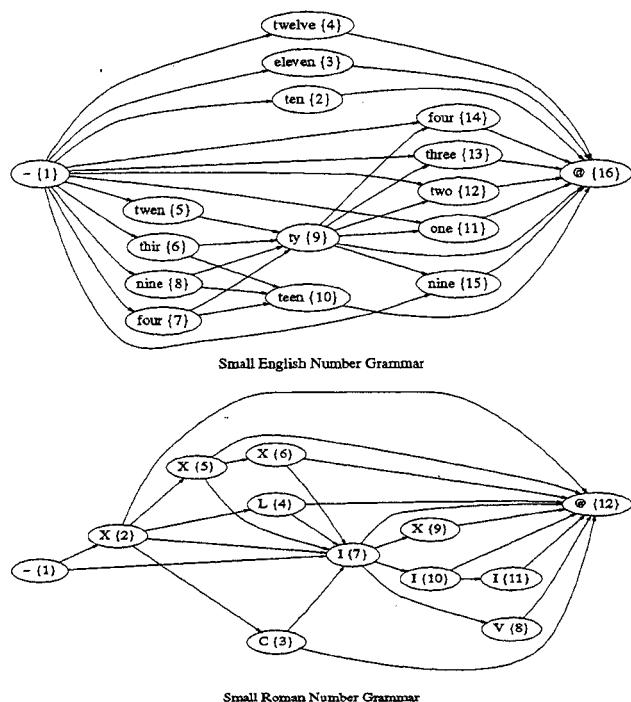


Fig. 3. Examples of Input and Output grammars for the translation of English numbers into Roman-numeral representation.

We can now summarize the proposed approach as follows:

A. Training. Let S be a set of pairs $(x,y) \in L(G') \times L(G)$ (transcribed sentences):

1. Obtain an appropriate *Input Language Model* (Stochastic Grammar G').
2. Obtain an appropriate *Output Language Model* (Stochastic Grammar G).
3. Estimate condition probability matrices $P(r'|r)$, $P(not\ r'|r)$, $\forall r'$ of G' , $\forall r$ of G .

An adequate choice for *step 1* is the use of the "Error Correcting Grammatical Inference" algorithm (ECGI) [10], applied to all x of S . For *step 2* we can use (a regular, stochastic approximation to) the *formal grammar* of the output (semantic) language, if given; alternatively, we can also use ECGI with all y of S . *Step 3*, finally, is just frequency counting, as previously mentioned. It should be noticed that if the adopted grammar inference technique is *incremental* (as ECGI) these three steps can be simultaneously applied to each training pair in a single incremental pass over the training data.

B. Decoding or "understanding". Let x be a (new) input (NL) sentence to be translated into an output (SL) string:

1. Obtain a derivation of x through G' , $DG'(x)$.
2. For each rule r of G
 - 2.1 Compute the probability, $P_D(r)$ of r to participate in an output derivation, given that the input derivation is $DG'(x)$.
 - 2.2 Combine (factor) $P_D(r)$ with the probability of r , $P(r|G)$, supplied by the Output Language model G .
3. Find a most likely derivation \hat{D} , through the output grammar G , according to the probabilities computed in step 2.
5. The decoding result \hat{y} is the sequence of output terminal symbols associated to the optimal derivation obtained in step 3.

Step 1 can be adequately performed through maximum-likelihood, error-correcting parsing, which can be easily implemented as a Viterbi-like procedure if G' is a regular grammar. *Step 2* consists of computing $P_D(r) \forall r$ of G , as indicated in (3). For the combination of these "derivation probabilities" with the *language model probabilities* $P(r|G)$ (*step 2.2*), a *weight* exponent may be useful to balance the relative impact of both contributions. *Steps 2 and 3* can be combined into a *Dynamic Programming Search through the graph of G*. Depending on the type of output grammar used, path-length normalization may be convenient to avoid search biases for shorter derivations. *Step 5*, finally, can also be trivially integrated with *steps 2 and 3*.

A final remark about the outlined approach is that all that has been said about grammar rules can also be said about "more elementary" items, such as non-terminals (or states in a regular grammar). In order to compare both possibilities, the experiments to be described in the next section were carried out using both (rule | rule) and (non-terminal | rule) association probabilities.

4. Experiments

A number of experiments were carried out for testing the capabilities of the new method here introduced. First, an artificial setting was considered in which English sentences specifying a small subset of numbers had to be translated into the representation of these numbers as *roman numerals* (see Fig. 3). This was a restricted version of a task considered in [7], in which many input-output non-sequential situations appear. In this case, the input and output grammars can be easily established by hand, or they can be learned using ECGI. In both cases, the application of the techniques described in Section 3 led to perfect results when complete training data were used, and gracefully degraded as relevant training items were removed from the training material.

A second experiment aimed at testing the capabilities of the new method with real data. For this purpose, a subset of spontaneous context-independent English queries to the ATIS database was selected from the ATIS corpus [6]. The semantics of each sentence was represented by "Pseudo English" (PE) commands (also called *win* or *wizard-input*) that are provided with the corpus. PE commands were

created by the annotators during the "wizard-of-Oz" acquisition of the corpus. The SQL code required to retrieve the reference answers associated to each English sentence can be automatically and unambiguously obtained from PE commands by a parser called NL-Parser [5].

From the whole ATIS corpus, a subset of 1146 "simple", (*English,PE*) pairs was selected. This selection was made by taking into account the lengths of the English sentences (≤ 20 words), as well as the "simplicity" of the corresponding PE commands (only commands that did not need the use of parenthesis were allowed). The resulting (*English,PE*) pairs underwent a rather conventional preprocessing treatment in which articles, inflections, etc., were removed from the English sentences, and a number of words such as city names, airline names, dates, etc., were replaced by generic non-terminals for these items. Some examples of input-output pairs used in our experiments before and after preprocessing are shown here below:

Examples of (English, PE) pairs:

- (LIST ALL DIRECT FLIGHTS FROM BOSTON TO DENVER,
List direct flights from Boston and to Denver)
- (I'D LIKE INFORMATION ON TWA FLIGHTS FROM
WASHINGTON TO PHILADELPHIA,
List flights from Washinton and to Philadelphia and TWA)
- (COULD YOU PLEASE GIVE ME INFORMATION
CONCERNING AMERICAN AIRLINES A FLIGHT FROM
WASHINTON DC TO PHILADELPHIA THE EARLIEST ONE
IN THE MORNING,
List earliest morning flights from Washing and to
Philadelphia and American)

(English, PE) pairs after Preprocessing:

- (LIST ALL DIRECT FLIGHT FROM <city.gram> to <*city.gram>,
List direct flights from <city_win.gram> and to <*city_win.gram>)
- (I'D LIKE INFORMATION ON <airline.gram> FLIGHT FROM
<city.gram> TO <*city.gram>,
List flights from <city_win.gram> and to <*city_win.gram>
and <airline_win.gram>)
- (COULD YOU PLEASE GIVE ME INFORMATION
CONCERNING <airline.gram> FLIGHT FROM <city.gram> TO
<*city.gram> EARLIEST ONE IN MORNING,
List earliest morning flights from <city_win.gram> and to
<*city_win.gram> and <airline_win.gram>)

A set of 1000 of these pairs was used for training; testing was performed on the remaining 146 English sentences. The results are shown in Table 4.1. The required input and output (regular) grammars were both automatically learned by ECGI. The resulting input (English) grammar had 922 states, 1182 rules, and 257 terminals. The output (PE) grammar had 190 states, 329 rules and 105 terminals. In both cases, the number of real terminals (and rules) had been significantly larger after expanding the non-terminals used for city names, etc.

TABLE 4.1
SUMMARY OF SEMANTIC DECODING RESULTS

Association Probabilities	Training-Set Errors	Test-Set Errors
(state rule)	5.2%	15.7%
(rule rule)	1.3%	11.6%

Although these results are rather limited, they clearly suggest the high potentiality of the here proposed approach to Language Understanding through Semantic Transduction.

6. Summary

Language Understanding can be seen as a process of translation between input natural language sentences and output sentences of an appropriate Semantic Language. Under this point of view, a new approach has been proposed to automatically learn the required transducers which overcomes certain problems of previous approaches with input-output sequentiality. We believe that this approach can be useful not only for Language Understanding itself, but in many other tasks of Language Processing, including Language Translation.

7. References

- [1] J.Berstel. "Transductions and Contxt-Free Langages". B. G. Teubner Stuggrt, 1979.
- [2] Brown,90: P.F.Brown et al.: "A Stocastical Approach to Machine Translation" Computacional Linguistics Volume 16, Number 2, 1990.
- [3] A.Castellanos, E.Vidal and J. Oncina: "Lenguaje Undestanding and Subsequential Tansducer Learning". First Int. Colloquium on Grammatical Inference: Theoty, Applications and Alternatives, Proc. Univer. of Essex, April 1993.
- [4] J.E.Diaz, A.J.Rubio, A.M.Peinado, E.Segarra, N.Prieto and F.Casacuberta. "Development of task oriented Spanish Speech Corpora" Eurospeech 93, Proc. 1993.
- [5] C.T.Hemphill, J.J.Godfrey, G.R.Doddington: "The ATIS Spoken LAnguage Systems, pilot Corpus". Proc. of 3rd DARPA Workshop on Speech and Natural Language, pp.102-108, Hidden Valley (PA), June 1990.
- [6] "MADCOW, Multi-Site DAta Collection for a Spoken Language Corpus". Proc. of Fifth Darpa Workshop on Speech and Natural Language, Harriman, NY, Feb 1992.
- [7] J.Oncina, P.Garcia and E.Vidal: "Learning Subsequential Transducers for Pattern Recognition Interpretation Tasks". IEEE Trans. on PAMI, Vol.15, No.5, pp.448-458, May 1993.
- [8] R.Pieraccini, E.Levin: "Stochastic Representation of Semantic Structure for Speech Understanding". EUROSPEECH-91. Proc. Vol.2, pp. 383-386. Génova. Sept. 1991.
- [9] R.Pieraccini, E.Levin, E.Vidal: "Learning How to understand LAnguage". EUROSPEECH-93.
- [10] N.Prieto and E.Vidal: "Learning Languages Model Through the ECGI method". EUROSPEECH-91. Proc. Vol.2, pp. 395-398, 1991. (also in Speech Communication 11, pp. 299-309, 1992
- [11] E.Segarra: "Una Aproximacion Inductiva a la Comprension del Discurso Continuo". Ph.D. Disertation, Universidad Politecnica de Valencia. 1993.