# Verb Instantiation by Concept Coherence and Refinement

**Abolfazl Fatholahzadeh**
Dept. of Computer Engr.
École Supérieure d'Électricité
57000 Metz, France
Abolfazl.Fatholahzadeh@supelec.fr

**H. Altay Güvenir**
Dept. of Computer Engr. and Info. Sci.
Bilkent University
06533 Ankara, Turkey
guvenir@cs.bilkent.edu.tr

## Abstract

This paper outlines a new approach for the verb instantiation problem in the translation of a narrative text using both the theory of event/state concept coherence and refinement with operators. Our hypothesis is that sentences of a narrative source text are conceptually coherent, and its appropriate translation should have the same concept coherence properties. The approach has been implemented and called TSGR (Target Sequence Generation by Refinement). TSGR transforms the input, a set of event/state concepts, into a plausible sequence using refinement and concept coherence in the target language. A strategy for disambiguation of each verb containing multiple senses is described. TSGR has been tested for verb instantiations in English to French and English to Turkish translations.

## 1 Introduction

Methods for automatically transforming a sequence of an input (see Figure 1) associated with a text according to its context of use have both scientific and practical interest. The scientific significance stems from the fact that its solution involves addressing three major problems: (1) representation of concepts, (2) performing model-driven learning, particularly foreign language, and (3) reasoning about different causal interpretations.

The translation of a narrative text, by nature, follows a logical chain of causal inference structured by the author. Therefore, the reader tries to capture the inter-relationships of mutually defining event/state terms in a paragraph. Under this assumption, translation can be viewed as a problem of sequence prediction. Consider the following example, called thereafter the *Lake-example*[1].

> A peasant was chopping a tree in the woods by the lake. He dropped his axe and it fell with a splash into the water. Quickly he dove into the lake ...

Here, verbs 'drop', 'chop', and 'hold' are concept coherent since they mutually define one another; a part of 'chopping wood' is 'holding an axe', and to have 'dropped something' one must have at first 'held it'.

The couples of (*drop*, *chop*), and (*chop*, *hold*) are called concept coherent in the *theory of event/state concept coherence* advocated by Alterman (Alterman 85). This theory has been applied to eight single paragraph-length samples of English texts by Alterman and has been used to answer questions and to produce summaries of these texts.

According to the theory of event/state concept coherence the representation of a narrative text can be generated by a process of matching text against a dictionary of concepts, which are related by a small set of relation-types, and using the organization of the concepts in the dictionary to organize the instances of event/state concepts which appear in the text. Event concept coherence, a property of the dictionary, is determined as a function of distance between concepts. Two terms are event concept coherent if there exists a path between two concepts in the dictionary. Concept coherence provides a representation for the similarity between two pieces of the text that use the same event/state concepts, but have different causal interpretations. In the Lake-example, the concept of 'dropping' disables the 'chopping', but for the text

> The peasant was chopping wood. When he finished, he dropped his axe.

it does not. In both cases, the terms 'dropped' and 'chopping' are connected via the concept 'hold'.

Nagao (Nagao 93; Nagao 89) proposed a method for dealing with the multiple interpretations of an utterance by abduction and dynamic preference in plan-based dialogue understanding. We propose an alternative based on the theory of event/state concept coherence and refinement.

Refinement (Güvenir & Ernst 90) is a method for generating subproblem goals from the goal of a given problem and finding a set of *relevant operators* such that the goal of each subproblem is easier to satisfy than the goal of the problem (Güvenir & Akman, 1992). Such a sequence of subgoals, called a *strategy*, can be used to apply a GPS-based search algorithm efficiently. GPS (General Problem Solver) methodology (Ernst & Newell, 1969) implements the problem solving technique called *means-ends analysis*, and is designed to work on state space problems.

One of the problems that a learner of a foreign language faces is to correctly guess the appropriate sense of a verb with multiple meanings. Usually, he finds the correct instantiation of the verb by making several hypotheses-verifications between the knowledge hidden in the text and the dictionary. This is essentially a *state space search* problem. Considering the problem of translation as a state space search enables us to concisely formalize it to be solved by our Refinement method.

TSGR (Target Sequence Generation by Refinement)

[1] Source: "The Peasant and the Waterman" (Protter 61) and (Alterman 85).

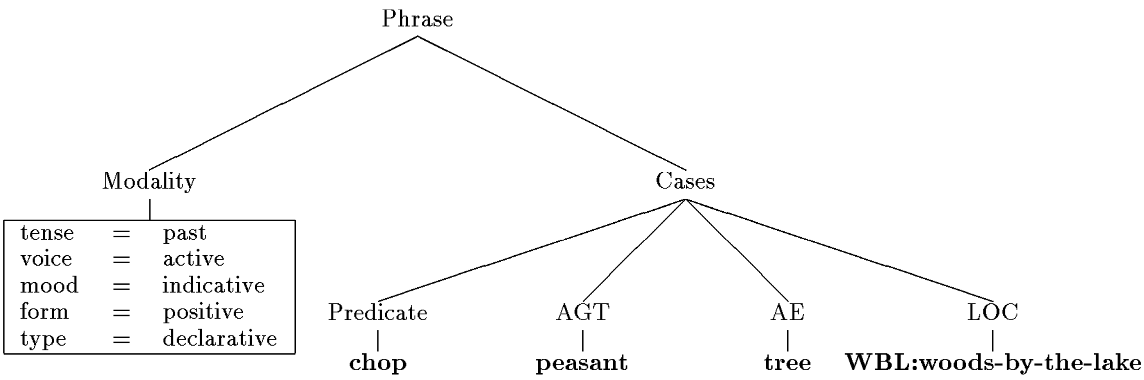


Figure 1: Semantic representation of the phrase 'A peasant was chopping a tree in the woods by the lake.'

tries to construct the representation of the target sequence on the basis of event concept coherence and refinement. For refinement purposes, what are *given*, in part, are those proximal concepts in the source dictionary. A pair of coherent concepts are connected to each other by an arc which indicates the relationship and its constraints. Among *known* information is the coherence relation that allows two non-proximal concepts to be connected, such as, 'falling' and 'diving'.

TSGR transforms the input associated to a text, by using a dictionary based on concept coherence, and by elaboration of *operators*. These operators will be used in the *refinement*, that allows us to determine the correct instantiation of the input verbs for the target language.

The next section describes how concept coherence can effectively be represented to be used in refinement for deriving a strategy. Then the problem of translation as a state space search problem is defined. Refinement is described in the following section. Finally, TSGR will be presented and explained in detail through the Lake-example.

## 2 Using Concept Coherence in Translation

We sketch the process of text conversion to case notation and introduce the notation that will be used in the rest of the paper.

### 2.1 Conversion to case notation

The method for obtaining the input to TSGR is the 'semantic representation' of the paragraph to be translated. This representation is based on the functional view of language which is partially inspired by the case grammar and follows the sublanguage theory advocated by Wilks (Wilks 76) and applied by other researchers (Deville & Paulussen 86). In this framework, the syntactic parser will derive a representation of the phrase according to the syntax of the sublanguage of the narrative text on hand. Initially, the syntactic component starts parsing the phrase. As soon as this component has information relevant enough to be exploited by the semantic parser, the second parser starts building the deep semantic structure of the phrase using syntactic and semantic information from three knowledge sources: the lexicon, the case grammar and the syntactic parser.

Table 1: Case attributes.

| Case | Abbr. | Description |
|---|---|---|
| AffectedEntity | AE | entity affected by an event |
| Agent | AGT | entity which instigates the action |
| Beneficiary | BEN | entity on which the event has a secondary effect |
| Destination | DES | location of a thing at the end of a motion |
| Instrument | INS | tool used in performing the action |
| Location | LOC | place where an event occurs |
| Object | OBJ | thing moved or transferred |
| Source | SRC | location of a thing at the beginning of a motion |
| Recipient | REC | receiver in a transfer of possession |
| StateOf | SOF | entity which the state describes |
| Theme | THM | event or a state embedded in a perception or communication |
| Time | TIM | time of an event |

The definition of case arguments that will be used in the next Section are given in Table 1. The cases are meant to account for the arguments that a particular sense of a concept can take. For example, the concept 'chop' includes 'an agent who performs the action', 'the entity affected by chopping' and optionally 'place where chopping occurs.' The semantic representation of the following phrase:

> A peasant was chopping a tree in the woods by the lake.

is presented in Figure 1, where a simplified version of the deep semantic structure is also given for clarity.

How is the assignment of the four words (*wood, by, the, lake*) to the 'location' information made? Although, the answer to the question is beyond the present study, we can say that, this information can be determined by the 'derivation rules' that are actually 'inverted Fillmorian transformations': starting from the syntactic function of a given noun phrase the rules will drive a case as a semantic function of that noun phrase. Then, the 'combining rules' e.g., $LOC1 = wood$ and $LOC2 = lake$, via a spatial preposition (such as 'by') can give the new information: $LOC$ = woods-by-the-lake. This rule can help to determine the exact meaning of the word 'by' in the target language. According to the English Webster dic-

Table 2: Concept coherence.

| Relation | Abbr. | Description |
|---|---|---|
| Class-subclass | sc | Property inheritance relation. |
| Sequence-subsequence | subseq | One event is part of another, and it occurs for a subinterval time. |
| Coordinate | coor | One event has parts that co-occur over the same time interval. |
| Antecedent | ante | One event must necessarily occur before another event. |
| Precedent | prec | One event with some regularity occurs before another event. |
| Consequent | conseq | One event always, necessarily, occurs immediately after the other. |
| Sequel | seq | One event follows another with some regularity. |

tionary, the word 'by' has fourteen properties.

The task of correct disambiguation of prepositions in analysis is far from being trivial. It looks like it is better to have an under-specification approach to sense distinction in analysis; however, one might want to further distinguish the spatial relation for generation. For that purpose, we have already developed a method based on reasoning with exact and approximate references, and the *likelihood* function (Fatholahzadeh 96).

### 2.2 Using Concept Coherence

TSGR assumes that the paragraph of a text to be translated has concept coherence among its verbs used in each sentence. Concept coherence is independent of both source and target languages. Alterman (1985) proposed 7 relations to relate two concepts. These relations are given in Table 2. There is one taxonomic relation: class/subclass (*sc*)[2]. Two relations are partonomics: sequence/subsequence (*subseq*), and coordinate (*coor*)[3]. Four of the relations are temporal: antecedent (*ante*), precedent (*prec*), consequent (*cons*), and sequel (*seq*)[4].

All knowledge about the relationships between two concepts in TSGR's dictionary is stored as a graph. The nodes of the graph represent the concepts, and the arcs represent the relations between concepts. Relations between nodes are stored in a quadruple, which has the following template:

```
[Relation Event/State1 Event/State2
          (Constraints)]
```

The first argument states the kind of relationship that exits between two concepts. The second and the third argument give the names of the two concepts being related, and the last argument is a list of constraints. The constraints specifies the required matches between the case arguments of the concepts.

```
[coor chop hold ((match AGT AGT)
                 (match INS OBJ))]
```

The relational form given above roughly states that there exists a coordinate relationship between chopping and holding. To establish this relationship, the instrument of 'chopping' must match the object of 'holding' and the agents of two concepts must match.

Here *match* is a function used to match the values of case arguments of two different instantiated event/state concepts. In order for the match succeed, its two arguments must be identical, or one must be a pronoun form of the other, or they must have a taxonomic relationship, or one of the arguments is empty.

The name of a concept is designated by its SL counterpart; so the event concept 'chopping' has the name *chop*. Since several event/state concepts can be mapped onto the same SL identifier, the names of concepts can include numbers and # signs to distinguish names; thus agent and agentless uses of 'moves' are represented by move2# and move1#, respectively. Furthermore, the meanings of these two concepts are differentiated: first, by their case arguments, and second by their relatives positions in the dictionary. Each concept's name is stored in the dictionary as a *node* of a graph. A proximal arc between two concepts is represented by one of the seven concept coherence relationships.

The instantiation of a concept is accomplished by matching the associated case notation of event against the dictionary. Given a phrase, such as the event

> A peasant was chopping a tree in the woods by the lake.

it is converted to case notation, input form for TSGR.

```
(chop  AGT peasant  AE tree
       LOC woods-by-the-lake)
```

The value of a concept is its name followed by a recognizer part due to the fact that several event/state concepts can be mapped onto the same source identifier. Each different sense of a concept is given a unique name, which is formed by the concept name followed by a number. Matching this representation against the source dictionary gives:

```
[chop1  ((AGT peasant) (AE tree)
        (LOC woods-by-the-lake))]
```

Since a verb may have several senses, there may be ambiguities to resolve. To determine the correct instantiation of the concepts in each sentence, one must take into consideration the different meanings. Table 3 shows 7 possible senses of 'chopping' concepts.

Assume that the dictionary used in translation contains 7 different meanings for the verb chop, 11 meanings for drop, 8 for fall and 3 for dive. In that case,

[2] For instance, a subclass of 'working' is 'chopping'.

[3] 'Travel' has three subsequences: 'depart', 'move', and 'arrive'. The corresponding event concepts between 'chopping' and 'holding' are in a coordinate relationship.

[4] An antecedent of 'dropping' is 'holding.' Sometimes before 'drinking' it is first necessary to open the container; then, a precedent of 'drinking' is 'opening'. A consequent of 'dropping' is 'falling'. Sometimes when two objects 'hit' one of them them 'breaks'. Then, in the event "the cup hit the floor and broke", the relationship between 'hit' and 'break" is sequel.

Table 3: Several senses of the verb 'chop'

| | Description | French | Turkish |
|---|---|---|---|
| 1 | *vi* to make a quick stroke or repeated strokes with a sharp instrument (*he has been chopping in the woods for an hour.*) | couper | kes |
| 2 | *vt* to cut into or sever by repeated blows of a sharp instrument (*he was chopping a tree in the woods.*) | couper | kes |
| 3 | *vi* to hit with a short downward stroke (*he chopped with his hand.*) | frapper | vur |
| 4 | *vt* to hit with a short downward stroke (*he chopped the ball with the club.*) | frapper | vur |
| 5 | *vt* to cut into bits, mince (*she chopped the meat with a robot.*) | hacher | kıy |
| 6 | *vi* to change direction (*the wind is chopping about.*) | change direction | yön değiştir |
| 7 | *vt* to reduce (*we chopped more than USD 1,000 off the budget.*) | baisser | azalt |

in order to instantiate these four verbs in the Lake-example, one must take into account 1848 possibilities for the disambiguation.[5]

The aim is to instantiate these four concepts such that all of these senses are concept coherent. According to our dictionary $chop_2$ and $drop_4$ are concept coherent, while the other senses are not.

The problem of determining the correct senses of the concepts such that they all are concept coherent in the whole of the paragraph can be defined as a state space search problem.

## 3 Translation as a State Space Search Problem

Translating a text into another language requires first determining the correct senses of each word to resolve the possible ambiguities. A narrative text usually follows a straight chain of event/state relations among its sentences. This coherence among the text can be used to resolve the possible ambiguities.

Ambiguities usually arise in determining the current sense of the concepts represented by verbs. For example, if we consider the verb 'chop', it has 7 possible senses. Each particular sense of a verb may have a different corresponding translation in the possible language. Determining the correct instantiations of the verbs in the target language can be formulated as a state space problem.

A state space search problem can be defined as $P = \langle I(s), G(s), S, O\rangle$. Here, $S$ is a *set of states* for the problem domain, $I(s)$ is the *initial statement*, $G(s)$ is the *goal statement* which specifies the goal states, and $O$ is the set of available operators. For instance, the goal statement in the Lake-Example is to determine the correct instantiation of four verbs, namely, 'chop', 'drop', 'fall', and 'dive' in the target language.

A problem instance can be given as $P_i = \langle P, s_i\rangle$, that is a problem $P$ along with an initial state $s_i$. A solution to $P_i$ is a sequence of operators $o_1, o_2, \ldots, o_k$ such that $o_i \in O$ and $o_k(o_{k-1}, \ldots, o_1(s_i)) \in G(s)$.

In order to formulate the translation of a text as a state space search problem, we represent a state as the *case structure* of a text with $n$ sentences. The *state* structure obtained from a case structure is shown in Figure 2.

A state structure is composed of state components, which represent the arguments of the case structure

[5] $7(chop) \times 11(drop) \times 8(fall) \times 3(dive) = 1848$.

$$
\begin{array}{llllll}
s : < & s_{1C}, & s_{11}, & s_{12}, & \ldots & s_{1m_1} \\
 & s_{2C}, & s_{21}, & s_{22}, & \ldots & s_{2m_2} \\
 & \ldots & & & & \\
 & s_{nC}, & s_{n1}, & s_{n2}, & \ldots & s_{nm_n} >
\end{array}
$$

Figure 2: State structure.

```
< [chop1 ((AGT peasant) (AE tree)
          (LOC woods-by-the-lake))]
  [drop1 ((AGT he)  (OBJ axe))]
  [fall1 ((OBJ axe) (DES water))],
  [dive1 ((AGT he)  (DES lake))]>
```

$$
\begin{array}{lllll}
s : < & s_{1C}, & s_{1A}, & s_{1E}, & s_{1L} \\
 & s_{2C}, & s_{2A}, & s_{2O} & \\
 & s_{3C}, & s_{3O}, & s_{3D} & \\
 & s_{4C}, & s_{4A}, & s_{4D} & >
\end{array}
$$

Figure 3: The case structure and the state structure of the Lake-example. Here, C, A, D, E, O, and L represent concept name, agent, destination, affected entity, object, and the location, respectively.

of the paragraph being translated. Here $s_{iC}$ is the concept and $s_{i1}$ through $s_{im_i}$ are case arguments of the $i$th sentence. As an example, the case structure of the Lake-example and corresponding state structure are illustrated in Figure 3.

An operator instantiates the values of the case arguments, by assigning a particular sense to a state component. The set of operators to be used in the translation is obtained by instantiating the templates given for each sense of a concept. An operator can be instantiated if the associated condition is satisfied by the case representation of the corresponding sentence. The conditions of an *operator template* specify the case arguments that *must* and *must not exist* in the case representation (Winston 84). For example, the template and its conditions for the second sense of 'chop' is shown in Figure 4. The operator to be used for assigning the second sense of the 'chop' action to the first sentence in the Lake-example is given in Figure 5.

## 4 Refinement

RWM (*Refinement With Macros*) is a strategy learning system for solving GPS-based state space search problems (Güvenir & Ernst 90). A *strategy* is defined as a decomposition of the problem into a sequence of easier

Must exist: AGT, AE
Must not exist: SRC, DES, OBJ, SOF

$$
\begin{aligned}
o_{<i>chop2}:\ (\quad & s_{<i>C} \leftarrow \text{chop2} \\
& s_{<i>A} \leftarrow Agent(CS_{<i>}) \\
& s_{<i>E} \leftarrow AffectedEntity(CS_{<i>}) \\
& s_{<i>L} \leftarrow Location(CS_{<i>}) \\
& \ldots)
\end{aligned}
$$

Figure 4: The template for the second sense of the 'chopping'. Here $< i >$ is to be replaced by the index of the sentence in the paragraph.

$$
\begin{aligned}
o_{1chop2}:\ (\quad & s_{1C} \leftarrow \text{chop2} \\
& s_{1A} \leftarrow \text{peasant} \\
& s_{1E} \leftarrow \text{tree} \\
& s_{1L} \leftarrow \text{woods by the lake})
\end{aligned}
$$

Figure 5: The operator to assign the second sense of the 'chopping' to the first sentence in the paragraph.

problems. Therefore, a strategy for solving a problem $P$ can be defined as a sequence of subproblems $P_1$, $P_2$, ..., $P_n$ such that solving them in sequence is equivalent to solving $P$, however each of the subproblems $P_i$ is easier than $P$. A GPS-based problem solver then solves each subproblem in sequence to obtain a solution to the given problem. TSGR uses the *refinement* technique of RWM to reduce the problem of translating a paragraph of text into a sequence of search problems where only a subset of operators are needed to establish a concept coherence relation between a pair of sentences.

In the RWM framework a set of states $Q(s)$ is represented by a *statement* $\mathcal{Q}(s)$ which is true for and only for the states $s \in Q(s)$. A statement is a set of atomic *statements*. An atomic statement is simply a binary relation between two state components, or between a state component and a constant value. A statement is interpreted as the conjunction of its atomic statements. An input problem to RWM is represented as a quadruple, $P = \langle \mathcal{I}(s), \mathcal{G}(s), S, O \rangle$. A GPS-based problem solver searches for a solution from an initial state $s_i$ satisfying $\mathcal{I}(s)$ to a state $s_g$ satisfying $\mathcal{G}(s)$ using the operators $o \in O$.

Formally, the refinement algorithm is based on a decomposition of the goal statement of a problem $P = < \mathcal{I}(s), \mathcal{G}(s), S, O >$ into subgoals by partitioning the goal statement $\mathcal{G}(s)$ into subgoals $\mathcal{G}_i(s)$, and finding the set of relevant operators $O_i$ for each subgoal such that $P_i = < \mathcal{I}_i(s), \mathcal{G}(s), S, O_i >$ satisfy the conditions: (1) $\mathcal{G}_0(s) = \mathcal{I}(s)$, (2) $\mathcal{G}_n(s) = \mathcal{G}(s)$, (3) $O_i \neq \emptyset$, for $1 \leq i \leq n$.

The formal definition of the refinement algorithm is given in (Güvenir & Ernst 90). The refinement algorithm is based on grouping those statements that have exactly the same set of relevant operators into subgoals. Then an ordering of these subgoals is searched such that the set of relevant moves for each subgoal is nonempty. Such an ordering of subgoals along with their relevant moves constitutes a strategy.

$$
\begin{aligned}
1.\ \mathcal{I}_1(s) &= \emptyset \\
O_1 &= \{o_{fall31},\ o_{fall32},\ o_{fall38},\ o_{dive41}\} \\
\mathcal{G}_1(s) &= \{(CC\ s_{3C}\ s_{4C})\} \\
2.\ \mathcal{I}_2(s) &= \{(CC\ s_{3C}\ s_{4C})\} \\
O_2 &= \{o_{chop12},\ o_{chop14},\ o_{chop15},\ o_{chop16},\ o_{chop17}\} \\
\mathcal{G}_2(s) &= \{(CC\ s_{1C}\ s_{3C}),\ (CC\ s_{1C}\ s_{4C})\} \\
3.\ \mathcal{I}_3(s) &= \{(CC\ s_{3C}\ s_{4C}),\ (CC\ s_{1C}\ s_{3C}),\ (CC\ s_{1C}\ s_{4C})\} \\
O_3 &= \{o_{drop22},\ o_{drop24},\ o_{drop25},\ o_{drop27},\ o_{drop28}\} \\
\mathcal{G}_3(s) &= \{(CC\ s_{1C}\ s_{2C}),\ (CC\ s_{2C}\ s_{3C}),\ (CC\ s_{2C}\ s_{4C})\}
\end{aligned}
$$

Figure 6: The strategy obtained by refinement for the Lake-example.

## 5 TSGR

TSGR uses refinement to effectively search for the correct instantiations of the concepts. A three stage strategy learned by the refinement algorithm for the Lake-example is given in Figure 6, where $CC$ designates the concept coherence relation.

Using this strategy, a GPS-based search algorithm can determine a set of values for the state components corresponding to the concepts. Initially, all state components representing concepts are set to their first senses. Therefore, the initial state $s_0$ for the lake-example is:

| | | | |
|---|---|---|---|
| $s_{1C}$ = chop1 | $s_{1A}$ = peasant | $s_{1E}$ = tree | $s_{1L}$ = WBL |
| $s_{2C}$ = drop1 | $s_{2A}$ = peasant | $s_{2O}$ = axe | |
| $s_{3C}$ = fall1 | $s_{3O}$ = axe | $s_{3D}$ = water | |
| $s_{4C}$ = dive1 | $s_{4A}$ = peasant | $s_{4D}$ = lake | |

The goal of the first stage is already satisfied by the initial state $s_0$. The subgoal $\mathcal{G}_1(s)$ is true because there exists a path between fall1 and dive1 in our dictionary. The portion of the dictionary relevant to the Lake-example is given in Figure 7.

1. [coor chop2 hold1
   ((match AGT AGT) (match INS OBJ))]
2. [ante drop4 hold1
   ((match AGT AGT) (match OBJ OBJ))]
3. [cons drop4 fall1
   ((match OBJ OBJ) (match DES DES))]
4. [sc fall1 descend1
   ((match OBJ OBJ) (match DES DES))]
5. [sc move1# descend1
   ((match OBJ OBJ) (match DES DES))]
6. [subseq travel3 move1
   ((match OBJ OBJ) (match DES DES))]
7. [sc travel3 travel2
   ((match OBJ OBJ) (match DES DES))]
8. [sc travel2 dive1
   ((match OBJ OBJ) (match DES DES))]

Figure 7: The portion of the dictionary relevant to the Lake-example.

In Figure 7, 'move1#' is used in the sense of an agentless 'move.' Travel2 is a specialized sense of 'travel3' that has the location as water.

The search program then proceeds to the second stage. The state $s_0$ does not satisfy the subgoal $\mathcal{G}_2(s)$, because 'chop1' and 'fall1' are not concept coherent in our dictionary. A search process is initiated using the operators in $O_2$. After applying the $o_{chop12}$ operator the state $s_1$ is obtained:

| | | | |
|---|---|---|---|
| $s_{1C}$ = chop2 | $s_{1A}$ = peasant | $s_{1E}$ = tree | $s_{1L}$ = WBL |
| $s_{2C}$ = drop1 | $s_{2A}$ = peasant | $s_{2O}$ = axe | |
| $s_{3C}$ = fall1 | $s_{3O}$ = axe | $s_{3D}$ = water | |
| $s_{4C}$ = dive1 | $s_{4A}$ = peasant | $s_{4D}$ = lake | |

Although $s_0$ and $s_1$ are similar, the first state represents the intransitive usage of 'chop', whereas the second state is for the transitive usage. Since the object is absent in the first phrase of Lake-example, the difference is evident. Otherwise, the value of $s_{1O}$ would have appeared in $s_1$. That is, the value of $s_{1O}$ of the first phrase is *nil* by default and it is omitted here. In $s_1$ the atomic statement ($CC\ s_{1C}\ s_{3C}$) is true because there exists a path between chop2 and fall1 via hold1 and drop4 as follows:

| | | |
|---|---|---|
| chop2 → hold1 | [coor | (match peasant peasant) (match axe axe)] |
| hold1 ← drop4 | [ante | (match peasant peasant) (match axe axe)] |
| drop4 → fall1 | [conseq | (match axe axe) (match water water)] |

By transitivity ($CC\ s_{1C}\ s_{4C}$) is also true. That is, the subgoal of the second stage is satisfied by the state $s_1$. The problem solver then proceeds to the last stage. Since the state $s_1$ does not satisfy the subgoal $\mathcal{G}_3(s)$, the operators in $O_3$ are applied. The application of $o_{drop24}$ to $s_1$ yields in the state $s_2$. Because the atomic statement ($CC\ s_{1C}\ s_{2C}$) is not satisfied, the state $s_2$ does not satisfy $\mathcal{G}_3(s)$. Therefore, the problem solver applies the next operator $o_{drop24}$ to $s_1$, which yields the state $s_3 = o_{drop24}(s_1)$:

| | | | |
|---|---|---|---|
| $s_{1C}$ = chop2 | $s_{1A}$ = peasant | $s_{1E}$ = tree | $s_{1L}$ = WBL |
| $s_{2C}$ = drop4 | $s_{2A}$ = peasant | $s_{2O}$ = axe | |
| $s_{3C}$ = fall1 | $s_{3O}$ = axe | $s_{3D}$ = water | |
| $s_{4C}$ = dive1 | $s_{4A}$ = peasant | $s_{4D}$ = lake | |

In $s_3$ the atomic statement ($CC\ s_{1C}\ s_{2C}$) is satisfied because there is a path between chop2 and drop4 in the dictionary. Similarly, the other atomic statements ($CC\ s_{2C}\ s_{3C}$) and ($CC\ s_{2C}\ s_{4C}$) of $G_3(s)$ are also satisfied by $s_3$; hence it is a goal state.

Although, for the Lake-example, there are 1848 possible assignments to concepts, TSGR, using the strategy derived by refinement, reaches the solution by generating only three states. The final step is to generate the sentences in the target language using $s_3$. For example, if the target language is French, then 'couper' will be used for 'chop.' The complete translation of the Lake-example in French will be:

> Un paysan coupait un arbre dans la forêt près du lac. Il lâcha sa hache et elle tomba dans l'eau en faisant flac. Rapidement, il a plongé dans le lac.

If the target language is Turkish, the verb 'kes' will be used for 'chop.' The complete translation will be:

> Bir rençper gölün yanındaki ormanda bir ağaçı kesiyordu. Baltasını düşürdü ve balta şapırtıyla suya düştü. O da hemen göle daldı.

Once the correct instantiation of verbs are determined, output of TSGR can be submitted to other natural language generation systems using a bilingual lexicon corresponding to the desired target language.

## Conclusions

We have outlined a new framework for the verb instantiation problem in the translation of a narrative text using both the theory of event/state concept coherence and refinement with operators. The program of this research, called TSGR, is for quickly and efficiently achieving accurate lexical choice for machine translation purposes. The present version of TSGR can determine the unique senses of the concepts used in the text by establishing concept coherence relationships between the verbs of each sentence.

The process of searching for the senses of concepts such that they all are concept coherent is a time consuming task. In order to make this search process more efficient, TSGR uses a *strategy* derived by the refinement algorithm. By experimentation, it is shown that using such a strategy improves the search process.

The complexity of the concept coherence theory is equal to the complexity of a bidirectional search on the graph representing the dictionary. The complexity of refinement has been studied elsewhere (Güvenir & Akman 92). The refinement method of TSGR allows us to reduce the search drastically compared to a pure brute-force approach. Our main goal in the present work was the study of verbsense disambiguation and measuring its complexity with respect to a narrative text.

TSGR is an attempt for being as an *intermediary, language-independent* representation which, in theory, separates the analytical side of a system from generative side, thereby suggesting a measure of modularity and complexity.

We plan to apply TSGR to (Güvenir & Tunç 96) for learning structural correspondences between two languages from a corpus of translated sentence pairs.

## References


(Alterman 85) R. Alterman. A dictionary based on concept coherence. *Artificial Intelligence*, 25(2):153–186, 1985.

(Deville & Paulussen 86) G. Deville and Hans Paulussen. A case grammar as an original linguistic model for the semantic representation of utterances in man-machine dialogue system. Universtaire Instellingen Anrwepen, September 1986.

(Ernst & Newell 69) G. W. Ernst and A. Newell. *GPS: A case study in generality and problem solving.* Academic Press, NY, 1969.

(Fatholahzadeh 96) A. Fatholahzadeh. *Reasoning with Exact and Approximate References in Scene Description.* Energy Information Management (volume IV) of Computer in Engineering (Book VI), pages 80–88. February 1996.

(Güvenir & Tunç 96) H.A. Güvenir and A. Tunç. *Corpus-Based Learning of Generalized Parse Tree Rules for Translation,* pages 121–132. *Advance in Artificial Intelligence*, LNAI, **1081**, Gordon McCalla (Ed.), Springer, 1996.

(Güvenir & Akman 92) H.A. Güvenir and V. Akman. Problem representation for refinement. *Minds and Machines*, 2:267–282, 1992.

(Güvenir & Ernst 90) H.A. Güvenir and W. Ernst. Learning problem solving strategies using refinement and macro generations. *Artificial Intelligence*, 44:209–243, 1990.

(Nagao 93) K. Nagao. Abduction and dynamic preference in plan-based dialogue understanding. In *IJCAI*, pages 1186–1192, 1993.

(Nagao 89) K. Nagao. Semantic interpretation based on the multi-world-model. 1989. In *IJCAI*, pp. 1467-1473, 1989.

(Protter 61) *A children's Treasury folk and Fairy Tale.* Channel, Great Neck, NY, 1961.

(Wilks 76) Y. Wilks. Natural language understanding system within the AI paradigm. Technical report, Stanford Univ., 1976.

(Winston 84) P.H. Winston. *Artificial Intelligence.* Reading Massachusetts. Addison-Wesely, 1984.