Internal Use Only (非公開)

TR-I-00179

Some Computational Applications of Lexical Functions

Ryo Stanwood† and Masami Suzuki

† stanwood@UHCCUX.BITNET

1990 . 8

Abstract

A semantic network is the proper context to fully realize the potential of Mel'cuk's system of lexical functions. Mel'cuk's system of lexical functions describe word relationships in such an explicit and systematic manner that their use in even simplest applications will improve the quality of the translation. These simple applications, however, crucially rely on the mechanism of synonymy. A semantic network allows us to avoid this problem. Unfortunately, the process of lexically realizing of this network requires more computationally sophisticated schemes. As a preliminary step, this paper suggests that during the course of translation the network can be managed through the use of partitions. Additionally, this paper proposes some constraints on processing algorithms to make such a network computationally more feasible.

ATR Interpreting Telephony Research Laboratories
ATR 自動翻訳電話研究所

Contents

1 Introduction

From a brief review of the literature, the author's general impression is that people are only recently discovering word collocations (the closest correlate to lexical functions) and their importance.[1]

This paper considers how Mel'cuk's system of lexical functions may be applied to machine translation. Mel'cuk's lexical functions are exactly the type of rich lexical information that is needed in machine translation. In this system, word-cooccurrence relationships are represented in such an explicit manner that they are ideal for computer implementation. The question, however, that has fascinated me is this: what sort of computational model uses lexical functions to its full potential? This paper is to be taken as a preliminary attempt to answer that question.

This paper will be divided into the following sections:

1.) An introduction to Mel'cuk's system of lexical functions will be given.

2.) This paper will demonstrate some straight forward applications of his system. The relative merits and problems of each method will be discussed.

3.) The use of a semantic network as the context for processing will be introduced and argued for. The problems associated with lexically realizing the network will be outlined. Furthermore, this paper will recommend the use of partitions and constraints on algorithms to meet this challenge.

4.) Finally, the results will be summarized

2 Mel'cuk's Lexical Functions

2.1 An Intuitive Introduction

In natural language, we often find certain words which seem to be naturally paired with each other. For example, in English, you *make* an excuse, but in Japanese, you literally *do* an excuse (iiwake o suru). There is no obvious semantic reason, why you can not *do* an excuse in English, it is just that English speakers never express the idea in that manner.

There are many other such pairs. To name a few: in English: we *suffer* loss, *recoup* losses, *undergo* change, *take* tests, in Japanese: sonshitsu wo *koomuru*, sonshitsu wo *umeawaseru*, henka wo *heru*, shiken wo *ukeru*, etc.

Mel'cuk's lexical functions explicitly and systematically describe these word cooccurrence relationships based on syntactic and semantic criteria. This is best illustrated through examples.

2.2 Illustrative examples

Synonymy (represented by the lexical function *Syn)* is the most obvious and probably the most sorely abused of all word relationships in machine translation. Mel'cuk refines this notion by differentiating between synonyms of greater scope, of less scope, and of intersecting scope, indicating this with the familiar subset, superset and intersection symbols of set-theory.[2]

Syn $\subset$ (plane) = aircraft
Syn $\supset$ (plane) = bomber
Syn $\cap$ (plane) = helicopter

The function *Bon(x)* yields a modifier whose meaning is "well or good".

Bon (see_{verb}) = clearly
Bon ($sleep_{verb}$) = soundly
Bon (conscience) = clear
Bon ($reward_{noun}$) = handsome, rich
Bon (opportunity) = golden

The function *Magn(x)* carries the meaning of "extreme" or "extremely".

Magn ($want_{verb}$) = badly, terribly
Magn (think) = hard
Magn ($trouble_{noun}$) = deep
Magn (suspicion) = nagging

Magn (sound) = loud

The function *Ver(x)* yields a modifier whose meaning is "veritable, genuine, or authentic".

Ver(suspicion) = well-grounded   Ver ($promise_{verb}$) = solemnly

Notice that the part of speech associated with these lexical functionsdepends on the argument to the function. Thus Bon(see) = clearly (an adverb), but Bon(reward) = handsome (an adjective).

The following chart demonstrates the contrasts between these functions:

X	Bon(X)	Magn(X)	Ver(X)
coincidence	happy	odd, strange	mere
attention	kind	close	due

A compound lexical function is analogous to a composite function in mathematics. Thus, AntiVer(sleep) = Anti(Ver(sleep))= restlessly.

AntiVer (resistance) =	token	**AntiVer** (love) =	unrequited
AntiMagn (sum) =	paltry	**AntiMagn** (crime) =	petty

The functions *S(x)*, *A(x)*, *V(x)*, *Adv(x)* yield a noun, adjective, verb, and adverb respectively. The subscript of 0 refers to the situation named by the function argument. Thus, these functions yield the syntactic derivatives of the situation named by the function's argument.

S_0 ($clean_{verb}$) =	cleaning	S_0 (young) =	youth
A_0 (beauty) =	beautiful	A_0 (see) =	visual
V_0 (registration) =	register	V_0 (before) =	precede
Adv_0 (peace) =	peacefully		

Notice again how lexical functions work across major syntactic category; both the noun "registration" and the preposition "before" may be input into the lexical function V_0 yielding a verb.

In Mel'cuk's system there are additionally an another important classof functions which serve to link situations with their actants. The functions *Oper*, *Func*, and *Labor* yield a class of semantically emptied verbs which Mel'cuk calls semi-auxiliaries. *Oper* specifies a verb in which the function argument(that is the situation name) occupies the object position. *Func* specifies averb in which the function argument occupies the subject position. *Labor* specifies a verb in which the function argument occupies the second complement position.

The numeric subscripts refer to the semantic actants of the situation named by the function argument.[3] These semantic actants fill the remaining vacant syntactic slots.

$\mathbf{Oper_1}$ (opposition)	=	offer	The rebels$_1$ offer *opposition.* (˜ to the army$_2$)
$\mathbf{Oper_2}$ (opposition)	=	encounter	The army$_2$ encountered *opposition.* (˜ from the rebels$_1$)
$\mathbf{Func_1}$(panic)	=	strikes	*Panic* struck the city$_1$.
$\mathbf{Labor_{12}}$ (suspense)	=	keep	John$_1$ kept Dave$_2$ in *suspense.*

Therefore, in these examples, $\mathbf{Oper_1}$ is what the first actant *does* to *opposition,* They *offer* *opposition.* $\mathbf{Oper_2}$ is what the second actant *does* to *opposition,* he *encounters* it. $\mathbf{Func_1}$ is what the situation (in this case, *panic)* *does* to its first actant. We say that *Panic* *struck the city. Finally, $\mathbf{Labor_{12}}$ is what the first participant *does* to the second actant *in* (or possibly *to*, *from*, *under*, etc.) the situation.

The next set of verbs are closely related. The syntactic alignment of *Real* is the same as that of *Oper.* In the same way, The syntactic alignment of *Fact* is the same as that of *Func,* it occupies the subject slot. Finally, the syntactic alignment of *Labreal* is identical to that of *Labor.* However, the difference between these two sets of functions is that *Real, Fact,* and *Labreal* contribute their own semantic content : the situation not only occurs, but is also "realized" or "consummated according to its intended design or purpose".

$\mathbf{Real_1}$(ambition) = fulfill $\mathbf{Real_2}$(request) = grant
$\mathbf{Fact_0}$(epidemic) = spread

Thus, in these examples: to realize an *ambition* is to fulfill an *ambition* where as to merely "have" an *ambition* is to harbor it. Likewise, when an *epidemic* spreads, it is working according to its design.

Finally, the functions *Incep, Fin,* and *Cont* are used to modify verbs to indicate their lexical aspect. These functions indicate whether the situation is starting, ending, or continuing, respectively.

$\mathbf{IncepOper_2}$(attention) = grab $\mathbf{ContOper_2}$(attention) = hold
$\mathbf{FinOper_2}$(attention) = lose

To indicate the end of the situation of "attention" by its second participant, we would use the verb "lose" and say "someone/something *lost* our *attention*".

2.3 Advantages of Lexical functions

Mel'cuk's system has the following advantages:

1) His system of lexical functions are so explicit and systematic that they are ideal for implementation on a computer.

2) They constitute a rich set which represent some twenty years of his research.

3) They can be supplemented. Because Mel'cuk's schema provides us with a non-arbitrary way of classifying word relationships, new functions are easily incorporated into the existing set as they are discovered.

4) They apply cross-linguistically. Not in the sense that every language has an identical inventory of lexical functions, but in the sense that a lexical function is the same for any language in which it occurs.

5) They greatly reduce the number of idioms which must be listed in our lexicon.

6) They allow us to focus our efforts on those semantic relationships holding across words which are pervasive. In lexical functions, we are not concerned with every possible semantic relationships which may hold between words, but only the main ones which cover large amounts of data.[4]

3 Example Applications of Lexical Functions

3.1 Lexical Functions as aRepresentation

Now that we have briefly introduced Mel'cuk's system of lexical functions, let us see how lexical functions may be used in translation. The first application uses lexical functions as representation for collocated words. This is useful when a direct word-to-word mapping is not possible.

Consider the following sentences:

English:	(the)battery died.	* (the)battery finished.	* (the)battery expired.
Japanese:	* batteri(ga) shinda.	batteri(ga) kireta.	

In this example, it is not possible to translate the English sentence, "the battery died" into its literal translation, "batteri ga shinda." Likewise, it is not possible to translate the sentence, the Japanese sentence "batteri ga kireta" into its literal English equivalents, "the battery expired" or "the battery finished."

When a battery dies, it ceases to function according to its intended purpose or design. Therefore, "die" can be represented as the $FinFact_0$ (battery) and "kireru" can be represented as the $FinFact_0$ (batteri). Thus, we represent "die" and "kireru" **indirectly** in terms of their relationship to battery.

Thus, our translation takes the following steps:

	(the) battery	died.
1)	(the) battery	$FinFact_0$(battery).
2)	batteri (ga)	$FinFact_0$(batteri).
3)	batteri (ga)	kireta.[5]

In this example, the process of translation takes three steps. The first step replaces the word "died" with its equivalent in terms of lexical functions. The second step, transfers those words in the source language which are directly mappable into the target language. The third step, evaluates the lexical function as a word in the target language.

In our next example, we will consider a more elaborate scheme.[6]

3.2 Lexical Function Paraphrases

There exist relationships between the lexical functions themselves which may be used for translation. For any given semantic structure, Mel'cuk's set of 60 paraphrasing rules give us all its deep syntactic structure paraphrases.[7]

The computational behavior of these paraphrase rules from the standpoint of generation has been studied in Boyer and Lapalme (1985) and also in Okamoto and Shimizu (1984). It seems that the significance of such work has not been fully appreciated.

In machine translation, these paraphrase rules allow us to translate a sentence, even when there exists an asymmetry between the lexical paradigms of the source and target language. In other words, the pairs of collocated words of both languages need not be related by the same exact lexical function. What is important is that lexical function in source language is paraphrasable in the target language.

Consider the sentence: "Kare wa tooroku shita".

Following the same steps used in the previous example:

1) Kare(wa) tooroku $Oper_1$(tooroku).
(replace "suru" with its functional equivalent.)[8]

2) He registration $Oper_1$(registration)
(transfer words which are directly transferable into English.)

But in English, there is no $Oper_1$(registration), so we can not perform the last step to evaluate the lexical function. There exists an asymmetry between the two languages, that is: there is a $Oper_1$ of tooroku in Japanese, but there is no $Oper_1$ of registration in English. If we were using our first method exclusively, we would have to stop here.

Now let us try using a paraphrase rule. One rule which might be applicable is the rule: $W = S_0(W)\ Oper_1(S_0(W))$. However, our input is not yet in the correct form for the rule to apply.

In English, we know that "registration" is a noun derived from the verb "register". (the S_0 of "register") Now, suppose we substitute "registration" with the lexical function S_0(register).

3) He S_0(register) $Oper_1$(S_0(register)

Now, our input is in the proper form for the paraphrase rule to apply. Applying our rule, that entire long string S_0(register) + $Oper_1(S_0$(register)) is replaced by the word "register".

4) He registered.

Thus, we see it is not necessary to have the same lexical function in both the source and target language.

3.3 Discussion of the Two Methods

Let us now discuss the advantages and disadvantages of both methods. The first method, where lexical functions are used as a representation for collocated word pairs, is the most elementary of the methods discussed. Despite this, the employment of even this simplest method would result in the improvement in the quality of many translation systems. Consider the following examples:

piano:play	piano:hiku
drum: play	doramu:tataku
loss: incur	sonshitsu:maneku
loss: recoup	sonshitsu: torikaesu

There is no obvious semantic reason, why in English, for example, we *recoup* *losses*, *recover* *losses* or *offset* *losses*, but we can not *regain*, *cease*, or *stop*, *losses*. Any system which can not describe such instances of restricted lexical cooccurrence, will not be able to handle similar pairs and therefore will not be able to deal with substantial portions of both the source and target language.

Despite this, there are rather major disadvantages to this approach. It assumes that when there is a collocated word pair in the source language related by a lexical function that there is also corresponding word pair in the target language such that:

1.) The function relating both the source language words and the target language words is the same exact function. Again, it can not deal with an asymmetry between the lexical paradigms of the source and the target language.
2.) The argument to the lexical function in the source and target language must be directly transferable. (i.e. strictly synonymous)

The second method, which uses lexical functions in paraphrase rules, has the advantage of being able to operate even when asymmetries exist between the target and source language's lexical paradigms. It does so by giving us all the possible paraphrases of a lexical function and its argument. In other words, it does not have the first disadvantage of method one.

But like the first method, its major disadvantage is that it is crucially driven by the strict synonymy of the arguments to the lexical function. If this synonym breaks down, the entire machine breaks down. It is this property that gives us problems. This suggests a rather different processing model is needed to appropriately use lexical functions.

4 The Semantic Network

4.1 Main Features of the Semantic Network

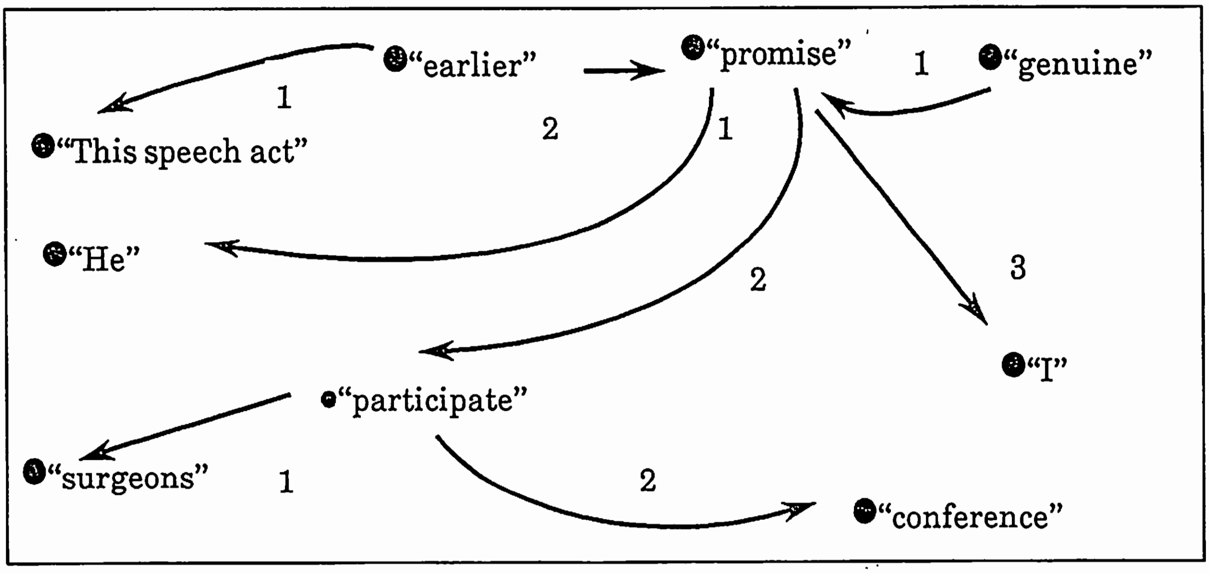

Fig. 1 Example of a semantic network

The context of this processing model is a semantic network. As such it is necessary for us to give a brief explanation of the network. In this example, this is a semantic network of the Meaning-Text variety. The length of this paper precludes any detailed discussion of either Meaning-Text theory or of this network. However, there are some important features of which should be pointed out:

1) The nodes of this network are either semantic primitives or composed of semantic primitives. We assume some universal set such as the set of 15 universal semantic primitives of Lingua Mentalis (Wierzbicka 1980).

2) Nodes are connected by unidirectional arrows indicating dependencies.

3) Arrows are labeled by the semantic actant numbers of the dependent nodes.

4) The network contains not only a part which is strictly isomorphic to the utterance, but also all the semantic assumptions of that utterance. For example, In the sentence: Yesterday, I took the train to Osaka and ate lunch there. It is assumed that I arrived in Osaka and stayed there long enough to eat lunch, although it is not explicitly stated.

5) This network is unordered to reflect the varying order of its possible realizations. Thus, this network may be realized as:

(A) He solemnly promised me that surgeons would participate in the conference.
(B) The participation of surgeons in the conference was solemnly promised tome.
(C) That conference participants would include surgeons was his solemn promise to me.
.... etc.

In the remaining sections, we will assume that the source language utterance has already been translated into such a semantic network.

This network is the context for our new processing model. Words cover parts of the network which are isomorphic to their definitions. Our new processing model uses lexical functions to also cover parts of our semantic network.

In our previous models, we had to consider the argument word to the lexical function in isolation. If that word was transferable, all other words contingent on it are also transferable, either directly or indirectly through paraphrase rules.

In this model, however, we can not consider a word in isolation, but rather we must consider the network as whole, collectively imposing constraints on the lexical realizations of its nodes.

For example, if the node labeled "promise" is lexically realized as the noun "promise", then the node labeled "He" may lexically realized as "his".9 In this example, the lexical function Ver covers the node labeled "genuine". The Ver of "promise" may be realized as either an adjective or as an adverb, depending on whether the node labeled 'promise' is realized as a noun or a verb. If the node labeled "promise" is realized as a noun, then Ver may be realized as an adjective.

In the context of a network, the value of a lexical function will be based on its argument word. The lexical realization of that word, however may be based on the lexical realization of another word and so forth. Unlike a word, however a lexical function may stand for several different possible lexical realizations in the text.

This example also demonstrates how the possible lexical realizations of a word may affect each other in the network. If that is true, then the order in which we process the network is significant.

4.2 Partitions of the network

In this section, we will deal with the problems arising in the lexicalization of that semantic network. By lexicalization, we mean the process which takes portions of the semantic network and turns them into words. We have already alluded to one of the problems: The problem of order of lexicalization. That is, as we have seen, the order in which we process the network will affect what lexicalizations are possible. The other major problem is the problem of compositionality. (i.e. what nodes or combination of nodes correspond to a word.) These problems suggest our previous model needs to be extended.

We will attack the problem of the lexicalization of that network into target language words in the following way: The essential strategy will be (1) to divide the network up into partitions and (2) the computationally manage these partitions during the course of translation.

The network is divided up into three partitions according to its communicative structure. The first partition, that we will consider will be that of Rheme verses Theme. The contrast between Theme and Rheme can be characterized as the difference between topic and comment, what is communicated about verses what is communicated. The Theme has the smallest amount of communicative dynamism. That is, it advances the process of communication the least. Whereas Rheme contains the highest amount of communicative dynamism. It advances the process of communication the most. In addition, A Rheme or a Theme may have an embedded Rheme or Theme of its own. Every part of the spoken utterance is either in Theme or Rheme of that utterance. The second partition which we will consider is the division between Old information and New information. Old information is that information which the speaker thinks that the hearer already knows about. Old information may not be lexically realized at all, but may simply reflect the semantic assumptions of the speaker. This is contrasted with New information. New information is the information which the speaker thinks is not totally predictable to the hearer. And finally, foregrounded and backgrounded information is distinguished. Something which is expressed by means of main predication is foregrounded. Something which is attributed is backgrounded.[10]

4.3 Partitions: Implication for Processing

Now let us see how these partitionings apply to the problems mentioned earlier. First of all, regarding the problem of order of lexicalization, the partition of Theme verses Rheme help to reduce the arbitrariness of our decision of where to start our lexicalization. Theme gives us the place to start and Rheme gives us the ending point.[11] Secondly, the partitions of Old verses New help us to answer the question of what part of the network is lexicalized. New information is that information which must be lexicalized. Old information is information which does not have to be lexicalized but could be. Finally, foregrounded verses backgrounded information help us to determine How a portion of a network may be realized. A foregrounded item is may be linked to other items by a main verb. A backgrounded item is likely to have no linking predicate. (e.g. an adjective in a noun phrase)

4.4 Translation Example -Managing Network

In this section, we will work through an example to showing how these partitions along with other principles may be used in translation.

At this point, we would like to suggest some constraints on algorithms processing the network. First of all, with regard to the problem of compositionality. Processing maybe performed using Grice's maxims of conversation (Grice 1967) as constraints. For example, Grice's maxim of manner states that an utterance should be orderly, brief, and to the point. What this means in network terms is that when we search for words to cover our semantic representation, we should choose those which cover the largest possible sections to keep our translation as concise as possible.[12] Grice's maxim of quantity states an utterance should be as informative as required for the purposes of the exchange. We could interpret this in the following way: the spoken utterance should contain all of New information and as little of Old information as possible.

Secondly, we begin our processing along the lines suggested by the source language. For example, in regard to the compositionality problem, we probably want to stay close to the composition indicated by the source language words. That is, we probably do not want to further decompose these nodes unless we fail to find lexicalizations in the target language under the original compositions. Likewise, when we attempt to translate this network into the target language, we may start according to the partitions implied in the source language. Initially, we can use the Theme and Rheme of the source language. However, the division of Old information verses New information should remain constant through out the translation.

Thirdly, we process this network deductively according to properties of the target language words. Thus, a target language words may differ in substantial ways from the source language. Syntactically, it may have a different arity with respect to the subcategorization in either the number of arguments or the syntactic features it subcategorizes for. Semantically, it may contain less information than the source language words or may contain more information than implied in the network. The amount of divergence tolerated will be controlled through the use of partitions.

Fourthly, when processing the network, we first cover dependent nodes with lexical functions where possible. This allows us to defer the lexicalizations of most dependent nodes, to first concentrate on lexicalizing those nodes which are relatively independent. The idea is to lexicalize your first order dependencies before moving on to your second order dependencies.

Now let us consider how partitions are used to computationally manage the network during the translation process.

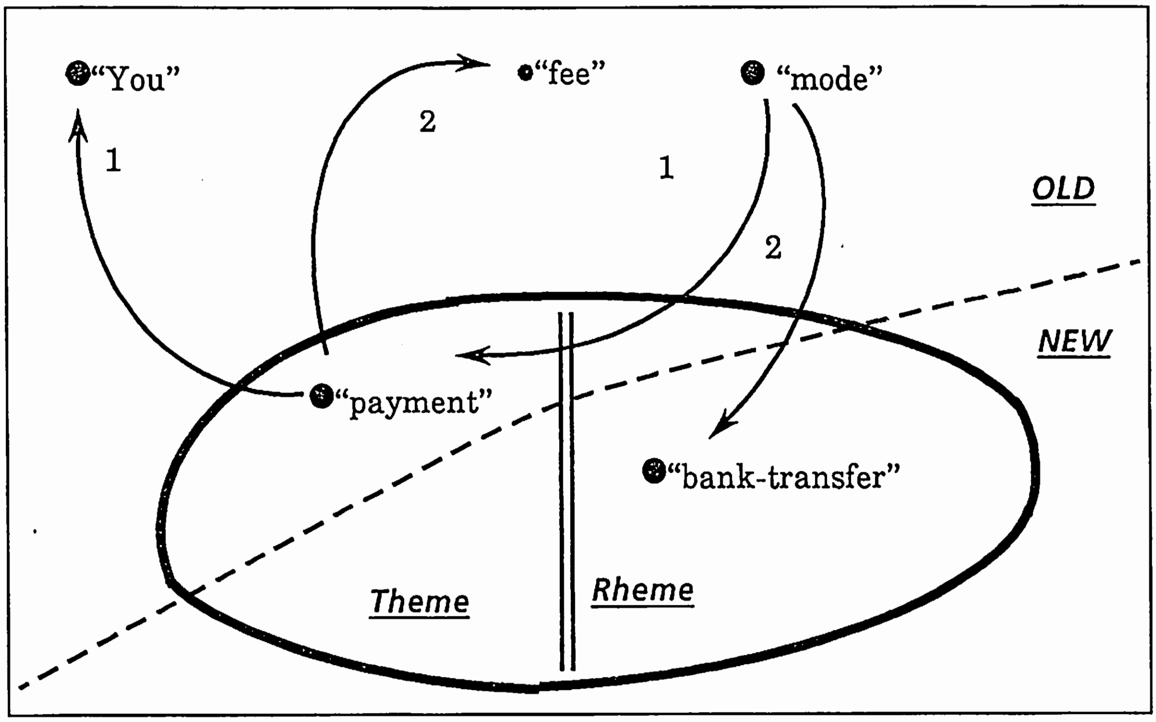

Fig. 2 A Partitioned semantic network

In this example, we have a division of Rheme and Theme. There is also a division of Old and New information. In this example: In Japanese, we say literally "payment is bank-transfer". The fact that "You" are making the payment and it is the payment of a "fee", and that bank-transfer is specifically the "mode" of payment is not covered by either the Rheme or Theme in Japanese. They are all part of the Old information cover. In English, however, the predicate "be" can not be used to relate bank-transfer as the mode of payment. In English, we must specifically state that bank transfer refers to the mode of payment.

This may be done by re-partitioning the network. Currently, the node "mode" is outside of both the Rheme and Theme. If we move it into the Theme portion of the network, the sentence may be realized as "The method of payment is bank-transfer". "Mode" is lexicalized as the English noun "method". It has one dependent which can be lexicalized as a prepositional phrase modifying that noun. If instead, we move "mode" into the Rheme division, it can be lexicalized as the preposition "by" relating bank-transfer to payment.[13]

There are also several alternatives to using the copula "be" to relate the situation "payment" to "bank-transfer". One way is to lexicalize the node labeled "payment" as the verb "pay". The verb "pay", however, like all English verbs, requires a subject. In this case, the node labeled "you" in Old information is moved into the Theme. As before, the node labeled "mode" is moved into the Rheme and lexicalized as "by". The resulting sentence is: "you pay by bank-transfer".

Recall that the lexical functions **Func**, **Labor**, and **Oper** link situations and actants. These verbs are semantically emptied and can be introduced into the network with negligible effects. The **Oper**$_1$ (payment) is "make". The verb "make" requires a subject which must be the first semantic actant in the situation "payment". Again, "you" must be retrieved from Old information and moved into the Theme. This sentence then becomes "You make payment by bank-transfer". The **Func**$_0$ (payment) is "is made". The **Func**$_0$ requires a subject which is the name of the situation. This sentence is then realized as "Payment is made by bank-transfer". This translation does not contain the element "You" from the Old information cover. Therefore, from the standpoint of Grice's maxim of quantity, this translation is better than the previous two.

As we process this network according to the properties of the target language words, we should notice that the translation often violates strict synonymy between the source language utterance and target language utterance. We should also notice that the network partitions might change.

What is an acceptable level of divergence? As an initial step, we suggest that the translation must unify (i.e. must contain no incompatible information with) the network as a whole. This constraint, however, probably needs to be weakened in some manner. As in unification, information which the speaker knows to be true would not be differentiated from information which he does not know to be false. Both would unify.[14]

As we can see, the idea of computationally managing this network is breaking new ground. The constraints and management techniques which we have proposed should be viewed as an initial attempt at something very ambitious.

5 Conclusion

In conclusion, we have found Mel'cuk's system of lexical functions is an explicit, systematic and largely untapped method of representing word relationships. Two simple applications in translation were suggested. Both methods fail when the arguments to the lexical functions are not strictly synonymous between the two languages. The real power of a lexical functions is realized in the context of a semantic network, where lexical functions serve as a cover for parts of that network. Lexically realizing this network, however, presents us with some problems. In an attempt to meet this challenge, we have proposed some methods to computationally manage this network. The network is partitioned according to its communicative structure. Constraints on processing algorithms have been proposed. Also importantly, the network is processed deductively according to the target language words. Under this model, strict synonym may be violated under controlled conditions.[15] A properly managed network of this variety will yield a translation which is both faithful to the communicative act and is pragmatically well-formed.

Acknowledgements

The authors would like to express our gratitude to ATR Interpreting Telephony Research Laboratories, especially to Dr. Kobei Habara, the chairman of the board and Dr. Akira Kurematsu, president, for their generous support of this project. During the course of our research, we have accumulated our share of intellectual debts. We would like to express our appreciation to the following people who have had a direct impact of the content of this paper: to Mr. Takahashi Toyoshima for his careful review of an earlier draft; to Mr. Ikuo Kudo and Mr. John Myers for their useful and practical comments; to Dr. Joseph Grimes of Cornell University who served as my theoretical lifeline during my stay at ATR, who took the time to respond thoughtfully and insightfully to my email questions despite his intense schedule; to the secretaries of the planning section for their responses to numerous questions about Japanese.

Bibliography

Boyer, Michel and Lapalme, Guy. 1985. Generating Paraphrases from Meaning-Text Semantic Networks. In: Computational Intelligence 1 (3-4): 103-107.

Grice, H.P. 1967. Logic and Conversation. Unpublished MS. of theWilliam James Lectures, Harvard University.

Grimes, Joseph. Forthcoming. Relations and Linkages in the Lexicon.

Grimes, Joseph. 1988. Information dependencies in lexical subentries. In: Relational Models of the Lexicon by Martha Evens, Cambridge University Press

Mel'cuk, Igor and Zholkovsky. 1984. Tolkovo-Kombinatoryj Slovar' Sovremennogo Russkogo Jazyka, Wiener Slawistischer Alamanch.

Mel'cuk, Igor and Pertsov, Nikolaj V. 1987. Surface Syntax of English. John Benjamin Publishing Company, Amsterdam/Philadephia

Okamoto, Tetsuya and Shimizu, Toshiaki. 1984. Japanese Paraphrasing System using Lexical Function, Transactions on Natural Language Processing, Information Processing Society of Japan, No.3, Vol.45, 1-8.

Wierzbicka, Anna. 1980. Lingua Mentalis, Academic Press.

Footnotes

1 The authors estimate that Mel'cuk's work is at least 15 years ahead of everyone else is doing in this area.

2 Grimes (1988) makes this distinction even finer-grained by including synonym discriminating information the lexical entries for his system. It seems that both schemes can be extended formally by viewing synonymy as simply unification, the labels of $\subset$, $\supset$, and $\cap$ as specifications of unificational behavior and function discriminating information as the information necessary for the subsumption of both definitions. Mel'cuk's subscripting of functions with the parts of word definitions, seems to just be the flip side of the coin of discriminating information. (i.e. It is the part of the word's definition over which the function holds. Discriminating information is that part of the word's definition over which the relation does not hold.) See Appendix B.

3 The semantic actant numbers simply correspond to theta roles.

4 It might be a good idea to clear up some possible misconceptions about lexical functions at this point. One paper, at least (Sakamoto and Ishikawa, date-unknown, Determining Japanese Case Frames from the Semantic Categories of a Verb) appears to have confused lexical functions with either case frames or subcategorization frames (for verbs, at least). The semantic actant numbers which correspond to theta roles are the closest correlate to case frames in this model. So, although lexical functions themselves are not case frames, we think of them as making reference to case frames.

Lexical functions have also been compared to conceptual dependencies. Within a conceptual dependency framework, lexical functions would correspond to clusters of conceptual dependencies. lexical functions are not intended to be primitive. My impression is that although conceptual dependencies are intended to be primitive, they seem to represent some rather complex semantic notions.

5 The careful reader will note that the transitivity of the relationship potentially plays an important role in the validity of the translation. Strictly speaking the relationship is directional. The value for a lexical function may not be unique. So while die $\Rightarrow$ $FinFact_0$(batteri) is true, the converse, ($FinFact_0$ (batteri) $\Rightarrow$ die), may not be. So the last step of this translation process is may not be strictly true. That is, when we evaluate the value of $FinFact_0$ (batteri) in Japanese, there might not be a unique value to that function. For purposes of translation, however, we could say that it makes little difference which of the alternative values we choose. Additionally, if we want to insure that the transitivity of the relation always holds, we could specify functional discriminating information for each of the possible values of a lexical function.

6 A demonstration program has been implemented in Lisp. This program is not as trivial as one might expect, since the program must access correspondingly

richer and more structured information. For more details, please refer to Appendix C.

7 There are two important points to mention here: First of all, these paraphrase rules are not necessarily entirely synonymous. Mel'cuk refers to them as "quasi-synonymous" operations. For most contexts, the difference in meaning is insignificant. For some contexts, such as quantification, however, this difference could well become significant. Secondly, in this paper, "Deep structure" refers the deep syntactic structure of Meaning-Text theory. It is roughly analogous to D-structure of Government and Binding Theory.

8 I am afraid that there is a minor flaw in this example. The verb "suru" takes on a wide variety of complements so it can not be said that "suru" and "tooroku" is an example of restricted lexical cooccurrence. This demonstrates a difference in the way languages encode meaning. English uses a specialized set of verbs to encode the meaning of **$Oper_1$**; whereas Japanese uses a transparent lexical means of encoding the same meaning. In other words, English uses a lexical function and Japanese uses a single word. (We could also say in Japanese, **$Oper_1$**(X)= "suru" for all X) Thus, in this example, "**$Oper_1$** (tooroku)" is derived from "suru" in a different manner than our previous example. The lexical function does not exist in Japanese, but the word ("suru") corresponds to (is "synonymous" with) the lexical function itself (**$Oper_1$**).

Although, the lexical function is derived differently, we believe the example itself is still valid and still demonstrates the main point which is that paraphrase rules may help us with such lexical asymmetries between languages.

It seems lexical functions sometimes can be map transparently into words or grammatical structures. The idea of mapping a lexical function in one language onto something other than a lexical function in another language needs further research.

9 The careful reader will note that the converse relationship is not true.

10 Note that the precise theoretical relationships between these partitions is not generally agreed upon by linguistics, however this does not preclude their use in a computational model. Initially, we may represent these partitions as independent. As we find more about their interrelationships, we may wish to constrain this representation.

11 We would also like to mention at this point that it seems that Theme also serves an additional function. It seems to be used to link the Old information cover with New information. That is, the speaker uses it to specify the part of Old information which the New information relates to. It is sort of a linkage, specifying how New information may be incorporated into the Old information cover.

12 Grice's maxim of manner also states that an utterance should avoid ambiguity. From the standpoint of the lexicon, avoiding ambiguity amounts to selecting words which are not homonymous with other words.

13 Consider our example: "Payment is by bank-transfer". Although this sentence seems to be well-formed, similar sentences do not seem to be well-formed:

* Bill is by check.	(shiharai wa kogite desu.)
* Eggs sale is by the dozen.	(tamago wa dasu-uri desu.)
* The letter is by express mail.	(tegami wa sokutatsu desu.)
* Touring is by bus.	(kanko wa bus desu.)

We are not sure how it is that "payment" seems to license the sentence in the first example, but the respective subjects of the following examples do not. Whatever that reason is, it means that the Japanese counterparts to these sentences can not be translated using the same method. Along with moving the node "mode" into the Rheme, something additional needs to be done to license that construction. For these constructions, it might be necessary to use one of the alternatives to the copula "be" which was discussed; that is, 1.) lexicalizing the subject as a verb. 2.) using a linking element such as Func or Oper.

14 It seems that Grice's maxim of Quality should be relevant here. This maxim states that an utterance should not contain something the speaker considers to be false nor what he lacks adequate evidence for. The first half of this maxim is in line with our suggested constraint. (i.e. the translation should not contradict the network.) In the second half, however, there is some ambiguity as to what exactly counts as "adequate" evidence and it means in network terms.

15 Mel'cuk (1987: 17-18) has a view on synonymy which is extremely relevant to our discussion. He believes it is important to differentiate between two types of synonymy: strict synonymy and synonymy for the purposes of the communicative act. Two utterances are strictly synonymous if no difference can be found. Two utterances are synonymous in the second sense, if no difference must be stated for the purpose of communication.

Appendix A : Theoretical Context

1 Purpose: To make the theoretical assumptions of this paper explicit.

2 An Important Definition:

Until now, a fundamental concept has not been defined: the lexeme, the object of description for the lexical function. In this appendix, we will adopt the traditional lexicographic definition of the term "lexeme". In the main body of this report, the term "word", the informal equivalent of "lexeme" was used. Although a lexeme normally corresponds to a word, there are some important differences.

A lexeme may be smaller than a word. The past tense "-ed" marker is such an example. The lexeme is that unit which underlies the words in an inflectional paradigm. For example, the lexeme "go" underlies the paradigm "go", "went", "going" and "gone".

The lexeme displays unique syntactic properties. Therefore, the noun "promise" and the verb "promise" are separate lexemes.

It displays idiosyncratic semantic properties, For example, we can not predict the meaning of the phrase "fly by night" from the meaning of its constituent words. Therefore, all idioms constitute single lexemes.

Finally, lexemes are distinguished on the basis of word sense. Therefore, "drink" in the sentence, "John drank some water", is to be differentiated from the "drink" in the sentence, "We are not allowed to drink at the office". The former refers to the drinking of any kind of liquid; the latter refers specifically to the drinking of alcohol. Although the two word senses are obviously related, they constitute two separate lexical entries.

In this appendix, we will use the term "word" and the term "lexeme" interchangeably to refer to the lexeme.

3. A Presentation of the Theoretical Framework

On Lexical Functions

In Grimes (forthcoming), lexical functions are defined formally in the following way:

> A lexical relation$_1$ is a binary function across the definitions of two words. ("binary" meaning that it is either true or false)

This is accompanied by a theory of word properties which simply states:

> All of a word's properties are derivable from its definition.

Such a view of word properties is useful for maintaining the theoretical independence of the lexicon. In the field of Linguistics, one is hard pressed to come up with "the indisputable winner" among competing grammatical frameworks. If in principle at least, all the feature/properties of a word that a particular grammatical framework requires are derivable from its definition, then the lexicon may avoid commitment to a particular syntactic theory.

If we combine these two statements, we find that lexical relationships may hold between any of a word's properties, not just by its syntactic and semantic properties, as done in Mel'cuk's work. For example, Grimes (1988) suggests lexical relations may used to relate words on the basis of the role which they play in inferencing. Another possibility is lexical functions may be used relate words by their meaning-dependent pragmatic presuppositions, such as those in Japanese honorification.

A Theory of Definitions

Numerous references have made to a lexeme's definition. It is therefore necessary to define what is meant by "definition". In particular, the view of Anna Wierzbicka is adopted.

A word's definition must follow two basic principles: 1) It must be substitutable for the meaning of the word in real language. It should be mentioned here that a word's definition is in prose. The only things which we can use to define words are words. A word is not defined by a bundle of artificial semantic features whose meaning must be inferred indirectly. 2) The word's definition must be composed of other words which are semantically less complex. This second principle is necessary for one to avoid the circularity in definitions rampant in almost all dictionaries. (The one notable exception to the rule is Mel'cuk's Explanatory Combinatorial Dictionary of Russian.) Notice, if one takes this idea to its logical conclusion, you will come to a level in which words can not be furthered defined. Words which can not be stated in terms of other words are primitive.

This is a very simple and intuitive notion, but the only person who has carried out this line of reasoning to its ultimate conclusion has been Anna Wierzbicka. The idea of formulating primitive tokens scares most people. Indeed, that was the first impression of the authors.[2] It seems when one formulates a primitive semantic primitive one is making a seems to be making a very strong statement the way human cognition takes place. The genius of Anna Wierzbicka's approach is that one can formulate a such a set void of such psycho-linguistic content. Furthermore, Wierzbicka has suggested that the set of primitives need not be a unique set. These primitives follow simply by virtue of the fact that language is a system which requires primitives as any other system.

In her current set, there are 15 primitives which she has been unable to further decompose for the last twenty five years. The set is:

1. I	5. something	9. not want	13. imagine
2. you	6. world	10. say	14.become
3. this	7. place	11. think of	15. be a part of
4. someone	8. want	12. know	

This set of primitives form a semantic metalanguage which is derived from natural language, but it differs from natural language in two crucial ways: both polysemy and homonymy have been eliminated.

Consider the set of sample definitions from Grimes (1987):

<u>a</u>$_5$: concerning$_1$ X
I say: I do not$_2$ imagine you can$_4$ think of X

<u>the</u>$_5$: concerning$_1$ X
I say: I imagine that you can$_4$ think of X.

<u>can</u>$_4$: concerning$_1$ X Ying
I say: me saying this is part of this world.
I am imagining this world becoming another$_3$ world,
part of which is X Ying.

<u>another</u>$_3$: not$_2$ this X

<u>not</u>$_2$: concerning$_1$ X
I say: thinking of X
I do not want you to not want to say: X Y's

<u>concerning</u>$_1$ X : I am thinking of X wanting you to think of X.

The numeric subscripts refer to the level at which the word is defined. For example, a_5, a level 5 word is defined exclusively in terms of primitives and words from level 4 and level 3. Thus, we see that a word is defined exclusively by words from a lower level. Level 1 words may defined only in terms of semantic primitives. This example demonstrates how circularity is avoided.

It might be good to mention here that although Mel'cuk is not yet willing to propose a set of primitives, he considers his approach to semantic decomposition to be "the direct result of Wierzbicka's pioneering work." (Mel'cuk and Polguere 1987) This list of "likely candidates" for semantic primitives appears in Mel'cuk and Pertsov (1987):

"(some)thing", "more", "say", "this speech act", "not", "set" (in the mathematical sense), "space", "time", "or".

Footnotes

1 The terms "lexical function" and "lexical relation" will also be used interchangeably.

2 Skepticism towards this set of primitives is quite understandable. Indeed, the gross failure of nearly all semantic systems may lead one to believe that no coherent semantic system can ever be developed. Before the reader dismisses this idea completely, we would like to invite the reader to two exercises. Wierzbicka's theory is **testable**. 1) If one wants to prove that a primitive is not a primitive, all one has to do is to decompose it. (i.e. state the same meaning in semantically more primitive terms.) 2) If one wants to argue that there are additional primitives, all one has to do is to find a word which can not be stated in terms of her current set.

Bibliography

Grimes, Joseph. 1988. Information dependencies in lexical subentries. In: Relational Models of the Lexicon by Martha Evens, Cambridge University Press

Grimes, Joseph. 1987. Pursuing the Undefinable. SIL Forum: July 6, 1987.

Grimes, Joseph. Forthcoming. Relations and Linkages in the Lexicon.

Mel'cuk, Igor A. and Pertsov, Nikolaj V. 1987. Surface Syntax of English.

Mel'cuk, Igor A. and Polguere, Alain. 1987. A Formal Lexicon in Meaning-Text Theory (Or How to do Lexica with Words), Computational Linguistics 1987, volume 13, Numbers 3-4

Mel'cuk, Igor and Zholkovsky 1984. Tolkovo-Kombinatoryj Slovar' Sovremennogo Russkogo Jazyka, Wiener Slawistischer Alamanch.

Wierzbicka, Anna. 1972. Semantic Primatives. Frankfurt: Athenaum Verlag.

Wierzbicka, Anna. 1980. Lingua Mentalis, Academic Press.

Wierzbicka, Anna. 1988. The Semantics of Grammar. Academic Press.

Appendix B : Examples of Lexical Functions

Purpose

The following list of examples has been compiled for the reader's reference. It was through working through such set of examples that convinced the authors of the importance of lexical functions to translation. The power and usefulness of Mel'cuk's system of lexical functions is best demonstrated through such illustrations. This list also offers a challenge to competing frameworks. In order to translate these instances of restricted lexical cooccurance, a system must have some means of unambiguously describing them. A competing framework, then must be able to at least represent these examples.

Sources

A number of sources were used to compile this list. Many examples were adopted from an English translation of the introduction to Mel'cuk's Explanatory Combinatory Theory of Russian. Some of the examples, however had to be omitted from this list, since they do not constitute instances of collocation in English. Another major source was his *Dictionnaire explicatif et combinatoire du francais contemporain*. These examples were adopted from French into English and Japanese (in part) entirely by the authors. A small number of English examples were adopted verbatim from two Mel'cuk's smaller works. In addition, a number of original examples worked out by the authors are included in this list.

Explanatory Notes

Footnotes are used to indicate an example's source. Any example which is not footnoted is an original example by the authors. It should also be mentioned that occasionally an argument to a lexical function is ambiguous. In those cases, we have subscripted the lexeme to identify its part of speech. A "V" and an "N" indicate a verb and a noun, respectively.

Simple Lexical Functions

(Melchuk, Igor and Zholkovsky 1984.)†1

Syn	shoot	fire	撃つ	発射する
Syn ⊂	shoot	machine-gun	撃つ	機銃
		shell		砲弾
$Conv_{21}$	include	belong	属する	含む
$Conv_{231}$	opinion	reputation	評判	意見
Anti	victory	defeat	勝利	敗北
Figur	fog	curtain	霧	カーテン
S_0	shoot	shooting	撃つ	射撃
S_1	teach	teacher	教える	教師
S_2	teach	pupil	教える	生徒
S_3	teach	subject	教える	科目
S_{instr}	shoot	firearm	撃つ	銃
S_{med}	shoot	ammunition	撃つ	弾薬
S_{mod}	examine	a point of view	試験	観点
S_{loc}	fight	battlefield	戦闘	戦場
S_{res}	learn	skills	学ぶ	技量
Sing	people	persons	人々	個人
Mult	ship	fleet	船	艦隊
Mult	student	student body	学生	学生組織
Cap	university	rector	大学	学長
Cap	faculty	dean	学部	学部長
Equip	theater	troupe	劇場	一座
Equip	hospital	personnel	病院	職員
Equip	marriage	spouses	結婚	新郎新婦
Centr	story	climax	物語	頂点
Centr	forest	thick	森	茂み
$Loc_{in}Centr$	desert	in the heart of	砂漠	～の中心
$Loc_{in}Centr$	road	in the middle of	道路	～の中央

A_2	shoot	under fire	撃つ	砲火を浴びて
Able_1	cry	tearful	泣く	涙ぐんだ
Able_2	doubt	doubtful	疑う	疑わしい
Qual_2	soil_v	white	汚す	潔白
Magn	examine	attentively	試験する	注意深く
AntiMagn	applause	thin	喝采	まばらな
AntiMagn	temperature	low	温度	低い
AntiMagn	losses	negligible	損失	無視してよい
Ver	surprise	unfeigned	驚き	偽りない
Ver	instrument	precise	道具	正確な
AntiVer	shame	false	羞恥心	
AntiVer	promise	false	約束	不誠実な
Bon	cut_v	clean	切る	きれいに
Bon	review	brilliant	評論	才気煥発な
Pos_2	review	positive	批評	肯定的な
Magn	temperature	high	温度	高い
AntiMagn	temperature	low	温度	低い
Pos_2	opinion	positive	意見	肯定的な
$\mathrm{AntiPos}_2$	opinion	negative	意見	否定的な
Loc_{in}	height	at		
Loc_{ad}	height	to		
Loc_{ab}	height	from		
Instr	typewriter	on		
Propt	fear	from (one's ~ for)		
Propt	love	out of (one's ~ for)		
Propt	experience	from		
Pred	next to	neighbor		
Oper_1	tears	shed		
Oper_2	resistance	run (into ~)		

$Oper_2$	arrest	fall (under ~)
		undergo
$Oper_1$	resistance	show
		put up
$IncepOper_1$	proposal	initiate
$Func_2$	proposal	concern
$Labor_{12}$	interrogation	subject (someone$_2$ to an ~)
$Labor_{32}$	lease	give (something$_2$ on ~)
$IncepOper_1$	popularity	acquire
$IncepOper_1$	despair	sink into
$IncepOper_1$	dive	go into
$IncepOper_2$	rule	fall (under the ~)
$ContOper_1$	influence	maintain
$ContOper_2$	influence	remain (under the ~)
$ContFunc_0$	odor	linger
$CausOper_1$	opinion	lead (someone$_1$ to an ~)
$CausOper_2$	operation	to put into
$CausOper_2$	slavery	force into
$CausOper_2$	control	to put under
$CausFunc_1$	hope	inspire (~ in someone$_1$)
		raise (~ in someone$_1$)
$CausOper_2$	dinner	prepare (something$_2$ for ~)
$CausFunc_0$	dinner	make
Liqu	illiteracy	wipeout
Liqu	campfire	extinguish
$Real_2$	hypothesis	confirm
$Fact_0$	knife	cut
$Real_1$	accusation	prove
$Real_2$	accusation	agree (with an ~)

$Real_2$	temptation	yield (to a ~)
		succumb (to a ~)
$AntiReal_2$	examination	flunk
$AntiReal_2$	advice	reject
$AntiReal_2$	application	turn down
$Fact_0$	hope	to come true
$Fact_1$	experiment	work out (for $someone_1$)
$Fact_2$	vessel	contain ($something_2$)
$Labreal_{12}$	gallows	string up ($someone_2$ on the ~)
$Labreal_{12}$	dinner	eat ($something_2$ for ~)
Sympt	fear,hair	(hair) to stand on end
$Prepar_1Oper_1$	dinner	appear (for ~)
$PreparOper_2$	dinner	serve ($something_2$ for ~)
$PreparFunc_1$	dinner	serve (~ to $somebody_1$)
$ProxOper_1$	despair	to be on the edge (of ~)
$ProxFunc_0$	thunderstorm	brew
Degrad	milk	go sour
Degrad	meat	go bad
Degrad	discipline	crumble
Son	dog	bark
Son	banknotes	rustle
Son	snow	crunch
Son	waterfall	roar
Imper	shoot	fire !
Imper	speak softly	shh !
Imper	take it	here !
Result	buy	have
Result	learn	know
		have skills

(Melchuk et al. 1984.)†2

Syn	hope	expectation	希望	期待
Syn ∩	help	assist	手伝う	助ける
		aid		——
$Conv_{21}$	in front	behind	前	後ろ
Anti ⊂	respect	contempt	尊敬	軽蔑
Anti ⊂	hope	despair	希望	絶望
Anti ∩	help	bother	手伝う	悩む
A_0	school	academic	学校	学問的な
		scholastic		学校の
A_0	dictionary	lexical	辞書	語彙的な
V_0	$promise_n$	commit	約束	～する
		$promise_v$		約束する
Adv_0	honest	honestly	正直	正直に
S_1	crime	criminal	罪	犯罪者
S_2	crime	victim	罪	犠牲者
S_1	fight	fighter	闘争	戦士
S_2	fight	opponent	闘争	相手
		adversary		——
S_{instr}	$paint_v$	brush	描く	ブラシ
S_{med}	$paint_v$	$paint_n$	描く	絵具
S_{loc}	fight	arena	闘争	闘技場
S_{mod}	write	handwriting	書く	筆跡
S_{mod}	$walk_v$	gait	歩く	足取り
		$walk_n$		歩み
		pace		歩調
S_{res}	fight	outcome	闘争	結果，成績
		result		成績

S_{res}	$duplicate_v$	$duplicate_n$	写す	写し
Sing	rice	grain	米	一粒
Sing	anger	attack	怒り	発作
Sing	anger	fit	怒り	発作
Sing	epilepsy	seizure	癲癇	発作
Sing	wind	gust	風	一陣(の風)
		blast		一吹き
Mult	wolf	pack	狼	群れ
Mult	bee	swarm	蜂	――
Mult	cranes	flock	つる	――
Mult	cow	herd	牛	――
Mult	cod	school	たら	――
A_1	search	missing	探索	行方不明の
$Able_1$	fear	scared	恐れ	怯えた
		fearful		恐ろしい
		frightened		怯えた
$Able_2$	fear	scary	恐れ	恐ろしい
		frightening		驚くべき
$Able_1$	hatred	malevolent	憎しみ	悪意ある
$Able_2$	hatred	obnoxious		不快な
		contemptible		卑しむべき
$Able_1$	anger	quick-tempered	怒り	短気な
		irritable		怒りっぽい
$Able_2$	admiration	dignified	賞賛	品位のある
		worthy		値する
Magn	noise	loud	騒音	うるさい
Magn	desire	ardent	願望	熱烈な
		burning		燃えるような
		fervent		熱烈な

Magn	promise	firmly	約束する	固く
Magn	need	pressing	必要	緊急の
		urgent		——
Magn	thank	deeply	感謝する	深く
		warmly		——
		whole-heartedly		——
Magn	satisfaction	great	満足	大きな
		deep		深い
Ver	wish	legitimate	願望	正当な
Ver	fear	justified	恐れ	正当な
Bon	indignation	righteous	憤慨	正当な　義憤
Bon	advice	valued	忠告	貴重な
Pos_2	opinion	favorable	意見	好意的な
		flattering		お世辞の
Loc_{in}	station	at		
Loc_{in}	(music)piece	during		
Loc_{ab}	personnel	within		
Loc_{in}	feudalism	under		
Loc_{ab}	gate	out from		
Loc_{ab}	morning	from		
Loc_{ab}	1970	since		
$Oper_1$	advice	offer	忠告	与える
$Oper_2$	advice	take		受け取る
$Oper_1$	operation	performs	手術	執行する
$Oper_2$	operation	undergoes		受ける
$Oper_1$	list	draw up	リスト	作成する
$Oper_2$	list	shows up on	リスト	記載してある
		appears on		記載してある
$Oper_1$	desire	experience	願望	経験

$Oper_2$	desire	arouse	願望	刺激する
$Oper_1$	question	pose	質問	する
		ask		——
$Oper_1$	housework	do	宿題	する
		perform		——
$Func_0$	wind	blows	風	吹く
$Func_1$	help	arrive	助け	届く
$Func_1$	hate	consume	憎悪	やつれさせる
$Func_2$	list	contains	リスト	含む
$Labor_{12}$	list	keep	リスト	記入する
		put down		——
Involv	wind	stir	風	ゆり動かす
		catch		捕らえる
Involv	blizzard	covered	大吹雪	覆われる
Involv	light	shines on	光	輝く
		illuminate		——
Involv	odor	permeate	匂い	充満する
		fill		——
$IncepOper_1$	form	take	形	とる
$IncepOper_1$	attack	commence	攻撃	始める
		launch		とりかかる
		trigger		仕掛ける
$FinOper_1$	influence	lose	影響	失う
$IncepFunc_0$	wind	stir up	風	巻き上がる
$IncepFunc_0$	difficulty	arise	困難	生じる
$IncepFunc_1$	fear	creep over	恐れ	ぞっとさせる
$IncepFunc_1$	desire	come over	願望	わき上がる
$FinFunc_0$	wind	still	風	静まる
$FinFunc_0$	fear	abate	恐れ	和らぐ

$FinFunc_0$	doubt	disappear	疑い	晴れる
		vanish		消える
$CausOper_1$	despair	drive to	絶望	押しやる
$CausFunc_0$	difficulty	arise	困難	生ずる
		manifest		明らかになる
$CausFunc_1$	doubt	sow	疑い	種を蒔く
		cast		投げかける
$CausFunc_1$	anger	provoke	怒り	引き起こす
		stir up		かき立てる
$LiquFunc_2$	attention	divert	注意	そらす
		distract		——
$LiquFunc_0$	doubt	remove	疑い	除く
$Liqu_1Func_0$	fear	subdue	恐れ	和らげる
		overcome		克服する
		conquer		——
$Liqu_1Func_0$	anger	stifle	怒り	押さえる
$LiquFunc_0$	difficulty	remove	困難	取り除く
$Real_1$	$promise_n$	fulfill	約束	果たす
		carry out		実行する
		keep		守る
$Real_1$	problem	resolve	問題	解決する
		settle		——
		sort out		区分けする
$Real_1$	anger	unload	怒り	発する
		vent		ぶちまける
$Real_1$	trap	trigger	罠	かける
		release		——
		spring		——
$Real_2$	trap	fall into	罠	はまる

		caught in		つかまる
$Real_2$	order	carry out	命令	実行する
		obey		従う
		comply with		追従する
$Real_2$	advice	follow	忠告	従う
		take		――
$Fact_0$	wish	is fulfilled	望み	叶う
		comes true		叶う
$Fact_0$	suspicion	is confirmed	嫌疑	固まる
		checks out		――
$Fact_2$	memory	is retained	記憶	保たれる
		remains		残る
$Labreal_{12}$	memory	keep	記憶	留める
Manif	joy	radiate	楽しみ	広がる
$Caus_1Manif$	excuse	make	言いわけ	する
		offer		――
$Caus_1Manif$	demand	formulate	要求	まとめる
		draw up		――
$Caus_1Manif$	opinion	voice	意見	声に出す
		express		表明する
$Caus_1Manif$	respect	show	敬意	示す
Degrad	wine	go bad	ワイン	悪くなる
Excess	heart	palpitate	心臓	脈打つ
Excess	motor	race	モーター	空転する
Son	dog	bark	犬	吠える
Son	cat	meow	猫	なく
Imper	aid	help!	助け	助けて!
Sympt	anger,	gnash	怒り	きしらせる
	teeth		歯	

Sympt	anger, eyes	bug out	怒り 目	目を剥く
Sympt	drowsiness, eyelid	grows heavy	眠気 瞼	重くなる
Sympt	drowsiness, head	nods	眠気 頭	こっくりする
AntiMagn	surprise	mild	驚き	穏やかな
AntiMagn	wind	gentle	そよ風	
AntiMagn	hope	faint	望み	微かな
		feeble		微かな
AntiVer	$fear_n$	morbid	恐れ	病的な
		unhealthy		——
		pathological		——
$AntiFact_1$	memory	fail	記憶	欠ける
$Caus_1Fact_0$	memory	search	記憶	たどる
		comb		ときほぐす

(Mel'Cuk, Igor A and Pertsov, Nikolaj V. 1987.)†3

S_0	despise	contempt
A_0	sun	solar
Magn	need	great
Magn	settled(area)	thickly
Magn	illustrate	vividly
Magn	belief	staunch
$Oper_1$	analysis	perform
$Oper_1$	attention	pay
$Oper_1$	favor	do
$Oper_2$	analysis	undergo
$Oper_2$	attention	receive
$Oper_2$	control	be under
$Real_1$	promise	keep
$Real_2$	attack	fall to
Son	cow	moo
Son	window-	jingle
	panes	rattle

(Mel'Chuk, Igor and Polguere, Alain. 1987.)†4

A_0	city	urban
A_1	surprise	surprised
A_2	surprise	surprising
$ContOper_1$	contact	remain (in ~with $someone_1$)
		stay (in ~with $someone_1$)
		keep (in ~with $someone_1$)
Contr	top	bottom
Contr	night	day

$Conv_{21}$	more	less
$Conv_{3214}$	sell	buy
Gener	anger	feeling (of ~)
Gener	pain	sensation
		feeling (of ~)
$Labor_{12}$	esteem	hold (someone$_2$ in high/low ~)
Liqu	meeting	adjourn
$Oper_1$	attention	pay
Magn	escape	narrow
Magn	bleed	profusely
S_0	honest	honesty
S_1	sell	vendor
S_2	sell	merchandise
S_3	sell	buyer
S_4	sell	price
Syn	calling	vocation
Syn $\subset$	respect	veneration
Syn $\supset$	keen	interested
Syn $\cap$	escape	break out
		run away

Own Examples

Syn ⊂	plane	aircraft	飛行機	航空機
Syn ⊃	plane	bomber	飛行機	爆撃機
Syn ∩	plane	helicopter	飛行機	ヘリコプター
Syn ⊃	novel	story	小説	物語
Syn ∩	story	ballad	物語	バラッド
Syn ⊂	story	mystery	物語	ミステリー
Sing	medication	dose	薬物	(薬の)一服
Sing	time	interval	時間	間隔
Bon	see_V	clearly	見る	はっきりと
Bon	$sleep_V$	soundly	眠る	ぐっすりと
Bon	$reward_n$	handsome	報酬	手厚い
Bon	opportunity	golden	機会	絶好の
Bon	attention	kind	注意	親切な
Bon	coincidence	happy	偶然	うれしい
Bon	appreciate	fully	享受する	十分に
Bon	congratulate	heartily	祝福する	心から
		warmly		暖かく
Bon	conscience	clear	意識	明確な
Magn	trouble	deep	問題	深い
Magn	suspicion	nagging	嫌疑	うるさい
Magn	opportunity	great	機会	良い
Magn	breathe	deeply	呼吸する	深く
Magn	$sleep_V$	heavily	眠る	深く
Magn	$promise_V$	firmly	約束する	固く
Magn	$want_V$	badly	欲しがる	ひどく
		terribly		すごく
Magn	loss	terrible	損失	非常な
Magn	sound	loud	音	大きな

Magn	attention	close	注意	細心の
Ver	attention	due	注意	当然の
Ver	$excuse_n$	reasonable	言いわけ	筋の通った
Ver	opportunity	solid	好機	充実した
Ver	appreciate	sincerely	感謝する	心から
Ver	promise	solemn	約束	厳粛な
Ver	suspicion	well grounded	疑い	根拠のある
AntiMagn	sum	paltry	合計	些細な
AntiMagn	crime	petty	罪	小さな
AntiMagn	amount	negligible	量	無視できる
AntiMagn	$sleep_v$	lightly	眠る	浅く
AntiMagn	know	hardly	知る	わずかに
AntiMagn	rain	light	小雨	
AntiVer	resistance	token	抵抗	名ばかりの
AntiVer	love	unrequited	愛	報わぬ-片思い
AntiVer	$sleep_v$	restlessly	眠る	休まれずに
AntiVer	argument	shallow	議論	狭い
AntiVer	excuse	feeble	言いわけ	弱い
		flimsy		見え透いた
Figur	snow	blanket	雪	一面に覆う物
S_0	fly_v	flying	飛ぶ	飛行
S_1	fly_v	pilot		パイロット
S_2	fly_v	passenger		旅客
S_3	fly_v	aircraft		航空機
S_1	$lease_v$	landlord	賃貸する	家主
S_2	$lease_v$	tenant		店子
S_3	$lease_v$	premises		家屋
S_4	$lease_v$	rent		家賃
S_0	fly	flying	飛ぶ	飛行

A_0	moon	lunar
A_0	see	visual
V_0	before	precede
Adv_0	peace	peacefully
A_1	love	fond (of someone)
A_2	love	beloved
S_1	war	ally
S_2	war	enemy
Adv_1	peace	peaceably
$Able_1$	love	affectionate
$Able_2$	love	amiable
$Able_1$	eat	hungry
$Able_2$	eat	delicious
$Able_1$	doubt	skeptical
$Able_2$	doubt	doubtful
$Able_1$	love	affectionate
$Able_2$	love	amiable
$Conv_{3214}$	rent	lease
$Oper_1$	opposition	offer
$Oper_1$	challenge	issue
$Oper_1$	loss	suffer
		sustain
		bear
$Oper_1$	ambition	have
		harbor
$Oper_1$	request	make
		have
$Oper_1$	suspicion	entertain
$Oper_1$	promise	make

$Oper_1$	test	administer
		give
$Oper_1$	change	undergo
$Oper_1$	vacation	take
$Oper_1$	aid	solicit
$Oper_1$	grudge	bear
$Oper_1$	damage	sustain
$Oper_1$	assistance	lend
$Oper_1$	opportunity	encounter
$Oper_2$	request	take
$Oper_2$	opposition	encounter
$Oper_2$	advice	receive
$Oper_2$	attention	receive
$Oper_2$	suspicion	falls under
$Oper_2$	test	take
$Oper_2$	assistance	receive
$Func_0$	darkness	falls
$Func_0$	opportunity	arises
		presents itself
$Func_1$	disaster	befell
$Func_1$	anger	consumes
$Func_1$	panic	strikes
$Func_2$	suspicion	falls on
$Labor_{12}$	suspense	keep (someone in ~)
$Labor_{12}$	suspicion	hold (someone$_2$ under~)
$IncepOper_2$	attention	grab
$IncepOper_2$	influence	fall (under ~)
$FinOper_2$	attention	lose
$FinOper_1$	loss	recover

		recoup
		offset
$FinFunc_0$	opportunity	slips away
$FinOper_1$	opportunity	loss
		miss
$ContOper_1$	attention	devote
$ContOper_2$	attention	hold
$Caus_1Func_2$	suspicion	places (someone$_2$ under ~)
$Caus_1Func_0$	loss	incur
$Caus_2Func_0$	attention	draw
		attract
		arouse
$CausFunc_1$	loss	inflict
$CausFunc_2$	attention	call
		direct (~ to something$_2$)
$PreparFunc_1$	medication	prescribe
$Real_1$	ambition	fulfill
$Real_1$	opportunity	seize
$Real_1$	piano	play
$Real_1$	medication	take
$Real_2$	request	grant
$Real_2$	test	pass
$Real_2$	challenge	accept
		take up
$Fact_0$	epidemic	spread
$Fact_0$	axe	chop
$AntiReal_1$	opportunity	for go
$AntiReal_2$	request	reject
$AntiReal_2$	attention	escape

$AntiReal_1$	opportunity	waste
		neglect
$AntiFact_0$	opportunity	slipped away
$CausFact_1$	medication	administer

Extensions in Mel'chuk's system

Lexical Functions of 3rd order or above

$IncepPredPlus^{refl}$	temperature	rise
$NonPermOper_2$	criticism	protect (someone from ~)†1
IncepPredMinus	joy	fads
		wanes
IncepPredMinus	wind	calms
		subsides
		quiets
IncepPredMinus	hatred	subsides
IncepPredMinus	speed	reduced
		cut
		decrease
$SingS_0AntiFact_1$	memory	gap
$A_1MagnManif$	joy	resounding
$A_2nonPerm_{1(1)}Manif$	joy	mute†2
$SingS_0Fact_1$	attention	span

Lexical Functions subscripted with definition parts:

indicating which part of a word's definition the lexical function operates on.

$Labreal_{12[keep]}$	memory	store
$Labreal_{12[output]}$	memory	retrieve
$Magn_{[losecontrol]}$	fear	panic†1
$Magn_{[consequences]}$	illness	serious
		grave†2
$Magn_{[exposure]}$	danger	pressing
$Magn_{[risk]}$	danger	grave

Lexical Functions superscripted with semantic labels:
making their meaning more precise

$Magn^{temp}$	experience	long
$Magn^{quant}$	experience	considerable†1
$Magn^{temp}$	writer	experienced
$Magn^{quant}$	writer	prolific

Lexical Functions superscripted for degree of fulfillment:
for example:
"I" may indicate realization on psychological level
"II" may indicate realization on physical level

$Real^{I}_{2}$	invitation	accept
$Real^{II}_{2}$	invitation	take up
$Prepar^{I}Fact_{0}$	revolver	load
$Prepar^{II}Fact_{0}$	revolver	cock†1

Fused Lexical Functions:
Both the "value" of the function and the "argument" to the function are combined into one word

Magn	rain	heavy // showers
Magn	delicious	very // finger-lickin' good †1

Configurational
The constituent compound functions are not syntactically related

A_1nonManif(hatred) + $Func_0$(hatred)	smolder
	simmer
AntiMagn(hope) + Figur(hope)	glimmer
	gleam
	ray†2
AntiMagn(doubt) + Figur(doubt)	shadow

Footnotes

†1 indicates the set of examples is from:
Mel'cuk, Igor and Zholkovsky 1984. Tolkovo-Kombinatoryj Slovar' Sovremennogo Russkogo Jazyka, Wiener Slawistischer Alamanch, "The Explanatory Combinatorial Dictionary" Translated by S. Hankwitz.

†2 indicates the set of examples is from:
Mel'cuk, Igor et al. 1984. Dictionnaire explicatif et combinatoire du francais contemporain (pp. 5 -13) Les Presses del'Université de Montréal

†3 indicates the set of examples is from:
Mel'cuk, Igor A. and Pertsov, Nikolaj V. 1987. Surface Syntax of English (pp 25-27) John Benjamin Publishing Company Amsterdam/ Philadephia

†4 indicates the set of examples is from:
Mel'cuk, Igor and Polguere, Alain. 1987. A Formal Lexicon in Meaning-Text Theory (Or How to do Lexica with Words), Computational Linguistics 1987, volume 13, Numbers 3-4.

Appendix C :

Documentation for demonstration program

This is the documentation for a simple program which uses lexical functions as a representation for restricted lexical cooccurence. As noted in the main body of this report, this program must have access to correspondingly richer information in order to recognize analogóus word pairs in the source and target language.

This program translates input structures from a source language into a target language. The input for this program is a feature structure. As a matter of convention, type names will be represented in capital letters.

Specification for Input Structure

Overall Structure:

```
((FORMS        a list of forms to be translated )
 (SYNTAX       forms grouped syntactically    )
 (SEMANTICS    forms grouped semantically     ))
```

Detailed Structured:

```
((FORMS

 (a list containing possible representations for form #1 )
 (a list containing possible representations for form #2 )
 (a list containing possible representations for form #3 )

    ...    ...    ...    ...    ...
    ...    ...    ...    ...    ...

 (a list containing possible representations for form #n ))

 (SYNTAX       forms grouped syntactically )
 (SEMANTICS    forms grouped semantically ))
```

Fully Detailed Structure:

Structure of FORMS type:

```
(FORMS
  ( (LEXEME language-name lexeme-name)
    (FUNCTOR-ARG
      (RELATION relation-name) (LEXEME language-name lexeme-name))

  alternate representation #3 for form #1
     ...          ...          ...          ...
     ...          ...          ...          ...
  alternate representation #n for form #1    )

  (a list containing possible representations for form #2 )
     ...     ...     ...          ...          ...  ...
     ...     ...     ...          ...          ...  ...
  (a list containing possible representations for form #N)  )
```

Structure of SYNTAX type:

```
(SYNTAX
  (a b (d (e f) g) h i j))
```

This grouping should reflect the syntactic dependencies of the forms. [a-j] are the index numbers to the form in FORMS (from 1 to N). The zeroth element in the list is the governor, the next is the first deep syntactic actant, and so forth.

Structure of SEMANTICS type:

```
(SEMANTICS
  (b d a (g c f) e j h i))
```

This grouping should reflect the semantic dependencies of the forms. [a-j] are the index numbers to the form in FORMS (from 1 to N). The zeroth element in the list is the situation name, the next is the first semantic actant, and so forth.

Format of the Lexicon

We assume an entry in the lexicon contains at.least one element of the type LEXEME and may contain zero or arbitrarily many elements of the type FUNCTOR-VALUE. If the lexicon contains additional types, they will be ignored by this program.

The lexicon may, thus take the following form:

```
(

    (LEXEME language-name lexeme-name)

    (FUNCTOR-VALUE
        (RELATION relation-name)
        (LEXEME language-name lexeme-name) )

      ...    ...    ...    ...
      ...    ...    ...    ...
      ...    ...    ...    ...

    (FUNCTOR-VALUE
        (RELATION relation-name)
        (LEXEME language-name lexeme-name) )

)
```

The type LEXEME identifies the language of the lexeme and its name (L_1). In our test lexicon, we use "WA" as the language name for Japanese and "EI" as the language name for English.

The type FUNCTOR-VALUE contains the name of the lexical function (R) and a pointer to *second* lexical entry (L_2) such that: ($R(L_2) = L_1$.)

How to Use Demo

A warning about this demo program. It is very un-user-friendly.

Step 1: Visit file "scratch-pad"

To access this file, you may use c-x c-f "~stanwood/DEMO/scratch-pad" on AS15. "scratch-pad" contains some sample setq's which may be altered and changed depending on what input you would like to specify.

Step 2: Load lisp-functions and initializations

This can be done by issuing a command "M-x load-file" and specifying: "~stanwood/DEMO/initialize.el"

Step 3: Build input and execute program

The sample setq's use the following the local work variables:
forms, syntactic-structure, semantic-structure, input-structure.

Normally, you should execute the setq's for **forms**, **syntactic-structure**, and **semantic-structure** first. Then build **input-structure** by concatenating **forms**, **syntactic-structure**, and **semantic-structure** into a single list. **input-structure** is then input into the main program.

In lisp interaction mode, the setq's may be evaluated by moving the cursor to the end of S-expression, and typing either an ESC c-x or a linefeed. An ESC c-x will place the results in the minibuffer and a linefeed will place the results in the current buffer ("scratch-pad")

The main program is called by evaluating the lisp expression:
(translate-structure input-structure)

SAMPLE RUN

This sample run was done in EMACS lisp interaction mode. The S-expressions were evaluated by inserting linefeeds after the lisp expressions.

<start of run>

```
(setq forms '(forms ((lexeme WA dasu)) ((lexeme WA shiken)) ((lexeme WA hito))))

(forms ((lexeme WA dasu)) ((lexeme WA shiken)) ((lexeme WA hito)))


(setq syntactic-structure '(1 3 2))

(1 3 2)


(setq semantic-structure '(2 3 nil))

(2 3 nil)


(setq input-structure (list forms syntactic-structure semantic-structure))

((forms ((lexeme WA dasu)) ((lexeme WA shiken)) ((lexeme WA hito))) (1 3 2) (2 3 nil))


(translate-structure input-structure)

((forms ((lexeme EI administer)) ((lexeme EI test)) ((lexeme EI person))) (1 3 2) (2 3 nil))
```

<end of run>