Field-Effect Natural Language Semantic Mapping

Stuart H. Rubin[1], Shu-Ching Chen[2] and Mei-Ling Shyu[3]

ABSTRACT

This paper addresses the problem of mapping natural language to its semantics. It presupposes that the input is in random (compressed) form and proceeds to detail a methodology for extracting the semantics from that normal form. The idea is to enumerate contextual cues and learn to associate those cues with meaning. The process is inherently fuzzy and for this reason is also inherently adaptive in nature.

It is shown that the influence of context on meaning grows exponentially with the length of a word sequence. This suggests that rule-based randomization plays a key role in rendering a field-effect natural language semantic mapping tractable. An example of rule-based randomization for semantic normalization is as follows. Suppose that two commands to a robot are deemed to be equivalent; namely, "Grasp and pick up the glass" and "Hold the cup and raise it". Their mutual normalization might then be, "Grab container. Lift container." Clearly, the randomization process can be effected by rules. Also, the normalized syntax makes the result of any semantic mapping process – such as detailed herein – more efficient.

A natural language front-end is described, which is designed to reduce the impedance mismatch between the human and the machine. Most significantly, the effective translation of natural language semantics is shown to critically depend on an accelerated capability for learning.

Keywords: Machine Learning, Natural Language, Randomization, Translation

1. INTRODUCTION

The theory of randomization was first published by Chaitin and Kolmogorov in 1975 [1]. Their work may be seen as a consequence of Gödel's Incompleteness Theorem [2] in that it shows that were it not for essential incompleteness, then a universal knowledge base could, in principle, be constructed – one that need employ no search other than referential search. Lin and Vitter [3] proved that learning must be domain-specific to be tractable. The fundamental need for domain-specific knowledge is in keeping with the Unsolvability of the Randomization Problem [4].

2. NATURAL LANGUAGE FRONT END

The capability to understand natural language (e.g., English) is critically dependent upon a capability to learn [5]. Indeed, insofar as language skills go, intelligence may be said to be a measure, not of language skills per se, but rather of the capability for their accelerated acquisition.

The task is to map an arbitrary statement in a natural language onto a finite set of semantics. One member of this set may be simply, "I don't know or understand to what you are referring." That is,

$$\{M \mid M(S) \rightarrow T, where \mid S \mid >> \mid T \mid\} \qquad (1)$$

In relation (1), *M* denotes a mapping function where *S* and *T* denote spaces of sentential semantics. The relevant question is, "What is *M*?". To answer this question, one must break it into two constituent parts:

1. The linguistic problem and
2. The contextual problem

The linguistic problem (1) pertains to the randomization and normalization of the supplied sentential form, *S*, as a prelude to mapping its semantics in step (2).

There are two categories of randomization:

1. Syntactic randomization and
2. Semantic randomization

Consider the two English queries:

1. Which person(s) might hijack the plane?
2. Did someone hijack the plane?

Here, the phrase "hijack the plane" provides an opportunity for syntactic randomization. The two queries are subsequently internally represented as:

1. Which person(s) might 001?
2. Did someone 001?
3. 001 $\longrightarrow$ hijack the plane

Note that syntactic randomization is properly recursively defined starting with the longest common phrases.

Semantic randomization is more difficult to achieve than is syntactic randomization. However, when properly done it allows for a far higher degree of randomization to be achieved. Sometimes it is clear from the usage that one word or phrase may be directly substituted for another (e.g., "Hi" for "Hello"). More often, a fuzzy relaxation may or may not be in order to achieve the desired randomization (e.g., Under what conditions can "Which person(s) might" substitute for "Did someone"). Other times, the substitution is specifically enjoined (e.g., Never substitute TRUE for FALSE).

The process of semantic randomization is further made difficult by the contextual problem. That is, sometimes it is

[1] SPAWAR SYSTEMS CENTER, 53560 Hull Street, San Diego, CA USA srubin@spawar.navy.mil.
[2] Distributed Multimedia Information System Laboratory, School of Computer Science, Florida International University, Miami, FL, USA, chens@cs.fiu.edu
[3] Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL, USA, shyu@miami.edu

proper to make the substitution and other times it is not. The differentiating factor here is the context in which the substitution might occur.

The key to solving the contextual problem in the context of semantic randomization is to make use of transitive relations. That is, if $A\Re B \wedge B\Re C \rightarrow A\Re C$. Thus, for example, if, "Which person(s) might 001?" is deemed to be semantically equivalent to, "Dangerous person 001" and, "Did someone 001" is likewise deemed to be semantically equivalent to this, then both queries may be equivalently replaced by, "Dangerous person 001". Indeed, it can be argued whether or not the semantics are equivalent. Suppose that it is subsequently found that they are not. In this case, the system can either replace the existing map, or acquire a new map having distinct context. In either case, learning is provided for.

To summarize, first the sentential forms are syntactically randomized. Then, they are semantically randomized and normalized through the use of acquired context-sensitive mapping rules. A knowledge acquisition capability implies truth maintenance as well as an ever-increasing capability for randomization. Note that in actuality, syntactic and semantic randomization are co-variant in the sense that the application of one enables the application of the other. Also, the transformational rule base is self-referential, which makes for a more complex mechanics than detailed herein.

The contextual problem assumes a fully randomized sentential form. This form represents a string of symbols as in 002 003 001. Syntactic and semantic redundancy have been compressed out of the string. This serves to minimize its length, which greatly facilitates tractability in the contextual problem operations. Moreover, the further use of normalization operations implies that sentential semantics can be extracted using nothing more than contextual sequencing; albeit, this is not a trivial problem. We will see that there is an elegant solution however. Consider the following two sentential forms. Note that they are not randomized or normalized so as not to detract from their readability; although, it is clear that a proper system will preprocess them as previously described. Let,

α : The dog bit the man.

β : The man bit the dog.

Consider the following sentence. The task is to map its semantics to α or β - whichever has a closer semantics.

ε : The man ate the hotdog.

To begin, we create a symbol hash table for unique identifiers. The ids can be case-insensitive for purposes of this illustration:

the	1
dog	2
bit	3
man	4
ate	5
hotdog	6

Associative memories construct indices based on feature vectors. For example, instead of retrieving an object by its feature number, an associative memory would retrieve the same object from its complete description, or even from an incomplete description.

One of the interesting features of natural language is that it allows sentences to be constructed that can be ordered from general to specific. For example:

> ***"Someone is going someplace next week."*** is more general than either ***"Harry is going someplace next week."*** or ***"Someone is going to New York next week";***

and these are all more general than, "*Harry is going to the Village on Tuesday*."

One question that arises is, "Should one use *n* words taken as ordered pairs, or perhaps $\binom{n}{r}, r = 1, n$ - i.e., the set of singletons, ordered pairs, ordered triplets, ... , the sentence itself?" There are several things to be considered before stating an answer to this question. First, can the ordered pairs, say $(i, j), (j, k)$ capture the semantics of the ordered triplet, (i, j, k) and so on? Clearly, the transitive property vies for the use of ordered pairs as a substitute for higher orderings. Next, one must consider the computational complexity of the associated set operations, which is a function of the number of objects in the set. Compare the complexity of $\binom{n}{2} = \frac{n(n-1)}{2}$, which is $O(n^2)$ and

$$\sum_{r=1}^{n} \binom{n}{r} = 2^n - 1 \text{, which is } O(2^n). \qquad (2)$$

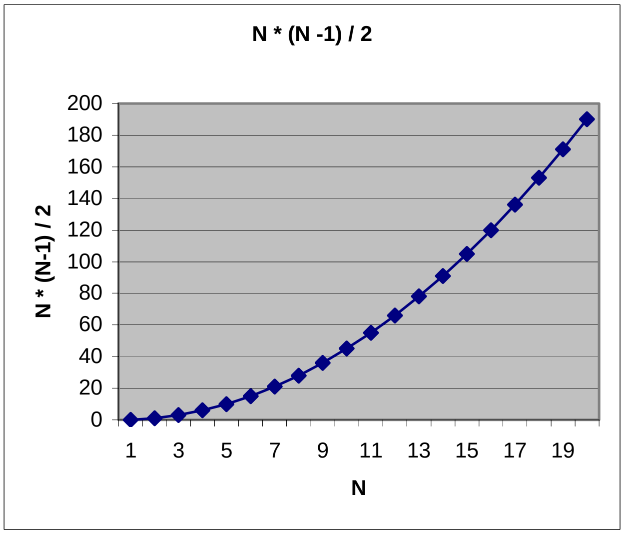

Fig. 1. Quadratic Complexity

It would appear in Fig. 2 that the human capability to comprehend context in a previously randomized sentence begins to fall off sharply somewhere between ten and twenty words. It is also noted that they system must be recursively decomposable to be capable of processing single words in a like manner to the processing of longer sentences. This suggests the use of the above $O(2^n)$ model in preference to the $O(n^2)$ one**.** To continue with our example, the sequential structure of the sentences is broken into sets of ordered pairs to characterize their underlying semantics. (The inclusion of ordered triplets and higher orderings would be similar.)

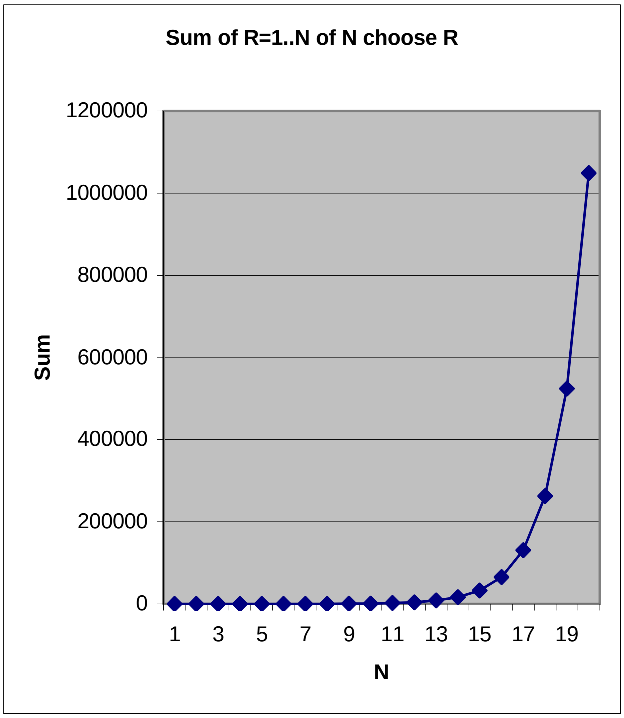

Fig. 2. Exponential Complexity

α : { (1,2) (1,3) (1,1) (1,4)
(2,3) (2,1) (2,4)
(3,1) (3,4) }

β : { (1,4) (1,3) (1,1) (1,2)
(4,3) (4,1) (4,2)
(3,1) (3,2) }

ε : { (1,4) (1,5) (1,1) (1,6)
(4,5) (4,1) (4,6)
(5,1) (5,6) }

Set intersection yields the possibility of a match. Here, possibility,

$$\rho \mid \rho = \frac{|\varphi_i \cap \varphi_j|}{|\varphi_j|}, \text{ such that, } |\varphi_i| \leq |\varphi_j|$$

For example:

$$\frac{|\alpha \cap \alpha|}{|\alpha|} = \frac{9}{9} = 1.0\,; \qquad (3)$$

a) Sentential semantics are reflexive.

$$\frac{|\alpha \cap \beta|}{\max\{|\alpha|,|\beta|\}} = \frac{|\beta \cap \alpha|}{\max\{|\beta|,|\alpha|\}} = \qquad (4)$$

$$\frac{|\{(1,2)\ (1,3)\ (1,1)\ (1,4)\ (3,1)\}|}{9} = \frac{5}{9} = 0.56$$

b) Sentential semantics are symmetric.

$$\frac{|\alpha \cap \varepsilon|}{\max\{|\alpha|,|\varepsilon|\}} = \frac{|\{(1,1)\ (1,4)\}|}{9} = \frac{2}{9} = 0.22$$

$$\frac{|\beta \cap \varepsilon|}{\max\{|\beta|,|\varepsilon|\}} = \frac{|\{(1,4)\ (1,1)\ (4,1)\}|}{9} = \frac{3}{9} = 0.33$$

$$\frac{|\{\alpha \cap \beta\} \cap \{\beta \cap \varepsilon\}|}{\max\{|\alpha|,|\beta|,|\varepsilon|\}} =$$

$$\frac{|\{(1,2)(1,3)(1,1)(1,4)(3,1)\} \cap \{(1,4)(1,1)(4,1)\}|}{9} =$$

$$\frac{|\{(1,1)\ (1,4)\}|}{9} = \frac{|\alpha \cap \varepsilon|}{|\varepsilon|} = \frac{2}{9} = 0.22 \qquad (5)$$

c) Sentential semantics are transitive.

- Sentence ε is more similar to sentence β than it is to sentence α.
- Sentence α is more similar to sentence β than it is to sentence ε.
- That is:

  a) The man ate the hotdog. ~ The man bit the dog.

  b) The dog bit the man. ~ The man bit the dog.

Clearly, (a) is correct; whereas, (b) is incorrect. It follows that if the language is sufficiently rich to be capable of self-referential statements, then no effective procedure can properly translate *all* of its semantics without error [4]. In other words, the proper translation of sentential semantics here provably requires the use of a *learning* algorithm. If in the case of (a), the oracle provides valid feedback to the effect that the two sentential semantics are indeed equivalent, then form two equivalent expanded sets to capture this equivalence:

$$\beta' = \varepsilon' = \beta \cup \varepsilon =$$
$$\{(1,4)(1,3)(1,1)(1,2)(4,3)(4,1)(4,2)(3,1)(3,2)(1,5)(1,6)$$
$$(4,5)(4,6)(5,1)(5,6)\}$$

Table 1. Quadratic vs. Exponential Complexity

n	$\binom{n}{2}$	$\sum_{r=1}^{n}\binom{n}{r}$
1	0	1
2	1	3
3	3	7
4	6	15
5	10	31
6	15	63
7	21	127
8	28	255
9	36	511
10	45	1,023
11	55	2,047
12	66	4,095
13	78	8,191
14	91	16,383
15	105	32,767
16	120	65,535
17	136	131,071
18	153	262,143
19	171	524,287
20	190	1,048,575

Version spaces imply search spaces. In view of this, an example follows, which demonstrates the creation of randomized identity transformations. Such identity transforms can be applied in transformational search. Again, it is clear that $\beta \equiv \varepsilon$. It follows that $\beta - \varepsilon \equiv \varepsilon - \beta$. It can be shown that this differencing does not change the semantics. Here,

$$\beta_2 = \beta - \varepsilon = \{(1,3)(1,2)(4,3)(4,2)(3,1)(3,2)\}$$

$$\varepsilon_2 = \varepsilon - \beta = \{(1,5)(1,6)(4,5)(4,6)(5,1)(5,6)\}$$

where, $\beta_2 \cap \varepsilon_2 = \varnothing$. Next, observe that $|\beta_2 \cup \varepsilon_2| \le |\beta \cup \varepsilon|$. It follows that identity transformations are randomizations.

Of course, sets cannot just aggregate. A pruning mechanism is also needed. Set differencing provides that needed mechanism. In the case of (b), the existing sets, α and β are updated:

$$|\alpha'| = \min\{|\alpha - \beta'|, |\alpha - \beta_2|\} = |\alpha - \beta'| = |\{(2,3)(2,1)(2,4)(3,4)\}|$$

$$|\beta_3| = \min\{|\beta' - \alpha|, |\beta_2 - \alpha|\} = |\beta_2 - \alpha| = |\{(4,3)(4,2)(3,2)\}|$$

α represented, “The dog bit the man.”
α' represents, “dog bit the man.”
β represented, “The man bit the dog.”
β_3 represents, “man bit dog”

Notice how the union and differencing operations converge to capture the concepts. We term this the *iterative randomization of sentential semantics*. This example works, but we expect to show that the concept can be scaled up to include many more and longer sentences and of course far more iterations than would be practical here.

An area for further research is the design of fuzzy sets, which would allow the sentential forms to be relaxed and have similar, rather than necessarily equivalent semantics. Clearly, the use of fuzzy logic means that set cardinalities would be real numbers, rather than integers. For example, if a number is used in a sentence, then fuzzy logic would have that number reduced to a qualifier (e.g., “The temperature was below 32 degrees Fahrenheit,” would become, “The temperature was below freezing.” Notice the use of context here.)

The mapping of many sentential forms to one implies the need to create new normalized forms that are mapped onto. Again, we refer to the normalized forms as *basis forms*. The need to create new basis forms follows from the *Unsolvability of the Randomization Problem*.

The proposed algorithm will employ *conversational echo*. For example, consider the following two distinct scenarios.

a) USER: What is the location of Americas most wanted person?
SYSTEM: Are you referring to a movie star?
USER: Most certainly not,
I am talking about known criminals.
SYSTEM: Are you referring to terrorists?
USER: That is correct.
...

b) USER: What is the location of Americas most wanted person?
SYSTEM: Are you referring to a movie star?
USER: Most certainly not,
I am talking about known criminals.
SYSTEM: Are you referring to deceased actors?
USER: Wrong again.
Computer, accept command set basis form.
...

In scenario (a), the system has succeeded in correctly mapping one of many possible sentential forms to a basis form. In scenario (b), the system did not succeed. True, it may or may not succeed given sufficient interaction, but that is irrelevant. The irrelevancy follows from the classic *Unsolvability of the Halting Problem*. In scenario (b), the user creates a new normal form to be the image of the semantic mapping function. Each normal form so created serves to form a mutually orthogonal or random set of basis forms. The definition of orthogonality here is complex, while the concept should not be.

It should be noted that the system’s echoed response can differ in structure or even in the base natural language used. This is because each echo is simply paired with the associated basis form. For example:

BASIS FORM: What time is it?
ECHO1: Would you like to know the time?
ECHO1': Est-ce que vous avez l’heure?
ECHO1'': Wieviel Uhr ist es?
...

Each basis form is paired with a computational semantics. These semantics must be effective procedures. A good approach is to slot the basis forms with SQL attributes and pair them with SQL queries. The SQL will retrieve the desired information from a relational database. The SQL should be an "orthogonal" or minimal instruction subset of SQL. In our experience, this will greatly ease writing the structured natural language to SQL translations.

Temporal databases must save information in relative, rather than absolute form to insure proper use of the set operations. For example, if a user asks for the time and the proper reply is that it is noon, then the implied reference is to a timing device and not to a stored absolute numerical value (i.e., noon).

Unlike a relational database, which stores records and uses rigid indexed-based searching, associative memories store associations representing the relationships of items in a particular context. Simple associative memories extract only co-occurrence relationships (i.e., remember the fact that two items were mentioned in the same context). Complex associative memories extract more advanced semantic relationships (e.g., remember the fact that two people live in San Diego).

A relational database could be used to search for fact-based answers such as, "How many people were bit by dogs last month?" An associative memory, in contrast with a relational database, can be used to answer questions such as, "What kind of people were bit by dogs last month?" and "What might cause a dog to bite someone?" The proposed approach has certain advantages for conversational learning. Consider:

CASE 1:

a) α is mapped against a database of normal forms.
b) The best match may echo a question, the purpose of which is to elicit further contextual information.
c) Let, β denote the user's reply to this query.
d) Then, $\alpha \cup \beta$ replaces α as the next iterate to be mapped against the database of normal forms.

CASE 2:

a) Again, α is mapped against a database of normal forms.
b) The best match echoes a question (answer), but the "power user" deems that this question (answer) is incorrect for any reason, where ε denotes the set corresponding to this basis normal form.
c) Then, $\varepsilon - \alpha$ replaces ε in the database of normal forms. (Note that $\varepsilon \cup \alpha$ replaces ε in the database of normal forms if the map is adjudicated to be correct. Notice that taken together, difference and union operations create a version space that attempts to converge upon the correct concept.)
d) New normal forms are appended to the database when appropriate. These normal forms may generate answers or questions. Answers are supplied initially. Incorrect answers are replaced with questions meant to acquire differential context. The use of questions serves to greatly increase the size of the mapped space (conversational learning).

If each recognized sentence is hashed to integer semantics, then a sequence of sentences (i.e., a paragraph) can be understood in the same manner that a sequence of words can be understood. Of course, larger constructs tend to be more random, which tends to delimit the utility of the fractal-based approach.

Randomization theory implies that the user need answer only the most novel of questions. All others may be iteratively answered by an interactive network of communicating domain-specific subsystems. Note that there is a plethora of evidence showing as Minsky puts it that the brain is a society of minds (i.e., functional areas). Note too that as the system learns that which is *novel* will evolve to higher and higher levels. *Intelligence* may be defined by the rate of this change.

Finally, success here will extend to impact computational vision. The connection between computational vision and natural language processing is dependent upon the development of appropriate context-sensitive concept representation languages for image processing.

3. CONCLUSION

Natural language understanding is fundamental to computing with words. It is necessary that we move beyond simple keyword techniques and into the realm of context-sensitive translation. This methodology will serve to reduce the impedance mismatch between the user and the machine. Moreover, it is now clear that any effective procedure for mapping natural language must, by definition, be adaptive.

4. ACKNOWLEDGMENTS

The author thanks SPAWAR Systems Center, San Diego for their financial support of this research. This work was produced by a U.S. government employee as part of his official duties and is not subject to copyright. It is approved for public release with an unlimited distribution.

5. REFERENCES

[1] G.J. Chaitin, "Randomness and Mathematical Proof," *Sci. Amer.*, vol. 232, no. 5, pp. 47-52, 1975.

[2] V.A. Uspenskii, *Gödel's Incompleteness Theorem*, Translated from Russian. Moscow: Ves Mir Publishers, 1987.

[3] J-H. Lin and J.S. Vitter, "Complexity Results on Learning by Neural Nets," *Mach. Learn.*, vol. 6, no. 3, pp. 211-230, 1991.

[4] S.H. Rubin, "Computing with words," *IEEE Trans. Syst. Man, Cybern.*, vol. 29, no. 4, pp. 518-524, 1999.

[5] S.H. Rubin, "New Knowledge for Old Using the Crystal Learning Lamp," *Proc. 1993 IEEE Int. Conf. Syst., Man, Cybern.*, pp. 119-124, 1993.