Revision 5 (final)
Approved By Dr. Fogarty
Dec. 1, 2005

Natural Language Processing:

uninteligent text processing

Jeff Plotkin
Math 370

Introduction

Natural Language Processing (or NLP) is the task of mechanically deriving the meanings of either spoken or written statements in natural languages. It is an area of computer science research, sometimes considered a subfield of artificial intelligence. It is also relevant to the field of linguistics and philosophy, all three of which are connected by the formal study of semantics natural languages [10].

Research in this area has given us technologies for grammar and spell check in word processors, foreign language translation, voice recognition, intelligent search and information retrieval and many others. The google search engine, through its use of highly sophisticated and innovative NLP systems has become the de facto web search engine in the world and coupled with the massive influx of formal information available on the web is quickly helping to eliminate the need for research in a physical library. Over the last 10 years huge amounts of information have been added to the world wide web which in turn has made google the first source for almost any research topic. Most people tend to even use google to find things which they have already found before and to spell check words that are not in the dictionary.

NLP research is primarily focused on the problems involved in programming computers to parse texts of natural forms of human written language. Parsing involves breaking a text apart into sentences, breaking those sentences apart into words and then determining what role each word plays in the sentence. For the "full glory" of NLP to be realized, as it has been in many science fiction works, a NLP system would have to be able to parse and compute the meaning (in some sense) of any text that a human can understand. However, such a goal requires an understanding of the answers to certain philosophical and psychological questions about human mental comprehension which are presently unanswered. Researchers in the department of computer science at the University of Massachusetts have been studying these problems carefully and working with scientists in the fields of psychology, neuro-biology and other areas to determine how the human brain actually works. Some of this research has already lead to advancement in the areas of artificial intelligence and machine learning.

Many useful things are possible with NLP, even without a proven theory of human consciousness. For instance, spelling and grammar check are standard on any commercial word processor and can work without any sort of artificial intelligence or special knowledge of the users

mental states. These text based checks do what they are supposed to do pretty well but they can also make errors in a few situations where either the system misses a grammar mistake or points out something that is not actually a mistake. More advanced systems based on AI and machine learning will avoid these mistake. The problem that generally arises is the complexity of the natural languages. They are like genes in animals: they evolve slowly as they are passed from generation to generation.

One interesting possible NLP technology that would in effect simulate intelligence does not require anything like a human brain, artificial intelligence or machine learning. Instead it may just require a databases of concepts defined in a tree structure. It would make use of as much computer hardware as is available given that the number of possible concepts expressed in a language is almost unlimited. This NLP technology will be comparable to what the calculator is to mathematics though. In as much as any text has meaning, the meaning of that text is computable, just like numeric functions are. The meaning of a statement is a function of its parts (which holds true deterministic and nondeterministic statements).

Functions

A function is a mapping from one set called the domain to another set called the range. If a member of the domain x is mapped to a member of the range y, then x must not be mapped to any other element other than y. [2]. In some functions every member of the domain and range is used, in others only a subset of one or both is used. Mathematical functions can easily be computed by a calculator because all possible inputs to the machine are in the domain and all possible outputs from it in range (for every function it can compute). But suppose we have a nondeterministic calculator for which there are two possible values for $f(k)$ for some f and constant k. If the calculator tried to compute $f(k)$, it would not be able to give the expression a value. It would get to a point where it has two choices and no rule for picking one of them. Deductively we know that if there are two values a and b such that $f(k) = a$ and $f(k) = b$ then a must equal b, because $f(k) = f(k)$. If not then $f(k)$ would not be equal to itself which a clear contradiction. [5]

The meaning of any function is computable if the meaning of the argument to it is computable and the function is defined for that argument. This 'meaning' is a denotational meaning. If $f(a) = b$ then from the perspective of denotational semantics [11], that $f(a)$ means b. For example, if someone says something like "Five plus three people will be at the event" they mean "Eight people will be at the event". The same is true for words in general because every word denotes something else such as physical object, events, property, conceptual object, system, etc.

For determining the meaning of a given text, if the text contains only one sentence, the meaning of the sentence is a function of the grammatical structure of that sentence and the component words. The meaning of any function based on numeric mathematics is automatically computable as the value of the function for that argument and the values of a non mathematical words is the value of a function of the values of the statements which comprise the definition. The meaning of the whole text is a function of the sequence of statements. In the situation where a statement has multiple interpretations, its value is a set of values and is said to be non deterministic. The number of possible interpretations of a text comprised of all non deterministic statements is exponentially more than the number of possible interpretations of any of the component statements.

The process of computing a value falls apart if there are ambiguities in the definitions of words just as a sequence of numerical computations falls apart if one of the computations involves division by zero. For instance, some words have multiple meanings independent of context in which the words appear which is an unrecoverable non deterministic situation. The underlying physical or mathematical meanings behind words should not ambiguous, therefore there should be some layer of a functional NLP that is never subject to ambiguity and that layer is the subject of the last sections below.

Languages

A language is "a system of signals, such as vocal sounds, gestures or written symbols which encodes and decodes information" [10,11]. Both naturally occurring and artificially constructed languages exist. Most artificial languages are formal languages, however Esperanto, Klingon and Elvish [11] are three informal languages which were constructed to seem like natural languages.

For a language to be useful it should be able to help its users accomplish goals and abstractly, this is done by the transfer a concept of some sort from the mind of one participant to another. In an optimal language, there would be no communication ambiguity aside from communication 'line noise' which is inherent on any communication channel using any physical medium. An optimal language is not likely to occur naturally, but those that occur naturally do work for the people that use them even if they aren't optimal.

Natural Languages

The rules of a natural language are empirical in that they are deduced over time from observation of usages of that language. The actual first origin of human natural language is currently unknown [10]. It is most likely that languages evolved along with animals that were able to use them. Many natural

languages currently in use in the world have been studied long enough that their rules are well defined.

The problem with natural languages is that even if the rules of a language are well defined, the people who use them may not conform to those rules. Some isolated groups may take on their own modified rules or they might not be aware of any rules at all. It is also possible for people to learn to speak and understand language just by experience, in which case their ability to conform to the rules of the language would vary with the speakers they had been exposed to. So it is hard to make a general purpose NLP system that will deal with all the possible special cases without turning to AI and machine learning techniques.

The rules for English and Latin are well known for instance. Using those rules it is possible to construct statements which can be decomposed systematically. In both Latin and English letters are the fundamental elements of the written language but words are really the fundamental objects of the language. Words are strung together to form sentences. Sentences generally convey three types of information: declarations, questions and commands. A declarative sentence or proposition takes on a value of either true or false. Questions, when uttered prompt the hearer to say something in response, i.e. the answer to the question which is directly related to the meaning of the question. A command, when uttered prompts the hearer to do something. Commands can be analyzed in terms of pre and post conditions to the command. Questions and commands can both be formally described by propositions, which is why we focus on propositions here.

All the Romance languages, as well as almost every European language follows a paradigm which can be called the subject-predicate paradigm. The subjects are nouns and the predicates are composed of verbs, nouns, adverbs, adjectives and prepositions, along with other possible word types [7,10]. The nouns other than the subject are generally called objects. Verbs and prepositions take objects for instance

Formal Languages

A formal language *L* is determined by a set of symbols called an alphabet, a set of syntax rules which state how the symbols of the alphabet can be combined to form well formed statements (or WFS) of the language the and semantic rules for computing the meaning of symbols in the language [5]. There are no exceptions to the rules in a formal language; a statement is either a WFS and has a value, or is not a WFS and has no value.

For any given string *s,* if *s* is a WFS then it is said to be in *L*. The language can also be given explicitly as a set of strings, but in general the set may be infinite. Since the rules of the language are

completely specified, a computer algorithm can be constructed to test strings for membership in the language [2] without the need for storing a potentially infinite set.

Formal languages are most commonly developed as logics or programming languages. A programming language is generally a language which humans can easily understand and which a computer program can translate into machine language (which is really not a language per se). A logic is a formal language with the addition of a set of valid argument forms. Numeric mathematics is a logic in effect. For example, Peano arithmetic can be used to do proofs regarding natural numbers and integers by recursively defining the natural numbers and induction rules.

Strangely enough, mathematics has traditionally been developed in much the way natural languages have, but all the facts of mathematics are logical. Gottlob Frege was the first to try to prove that mathematics as a whole was a formal language. Unfortunately he failed due to a number of paradoxes that arose in his system, one of which Bertrand Russell is famous for. Later Russell tried to create his own mathematical logic which had significant limitations (but no paradoxes). Years later Zermelo and Frankle succeeded to create a system to prove that mathematics has a logical foundation by creating a paradox-free logic in which all the necessary properties of numeric mathematics could be derived. Another system NGB, named for its esteemed creators can make the same claim. The systems were quite complex though, and that was not what most philosophers had hoped [6]. But they need not be used by mathematicians to prove mathematical facts. They are only used by computer scientists who want a very general model for theorem proving on a computer and mathematical logicians.

Meaning

Some concepts which people wish to convey can not be defined in a strictly scientific, mathematical or logical manner (or by SML for short). Some concepts have meaning in some sense, but that meaning varies from person to person and there may not be a "correct" interpretation (at least not yet). Consider the six main classes of words quoted from Roget's Thesaurus [12]:

I. Words Expressing Abstract Relations
II. Words Relating To Space
III. Words Relating To Matter
IV. Words Relating To The Intellectual Faculties; Formation and Communication of Ideas
V. Words Relating To The Voluntary Powers; Individual And Inter social Volition
VI. Words Relating To The Sentiment and Moral Powers

In an academic sense, SML defines all words of type II and III and in most cases words of type I. Some

words of type IV and V may not be fully definable by SML, yet if ever. Words of type VI are probably not defined by SML at all because even though we could use mathematics or logic to define words regarding morality for instance, they may be philosophical concepts that even philosophers do not agree on. For instanced words like ‘right’, ‘wrong’, ‘good’, ‘bad’, ‘evil’, ‘god’, ‘happiness’ are all words which are deeply personal in nature. Each person has a different idea of what they mean and not many people agree on an objective meaning.

How simple NLP works

For a text based NLP that works with the English language or any other well known language, information about the grammar of the language can be taken from a grammar text book. The information there can be used to construct a set of grammatical rules that the computer can recognize. The full list of words from an English dictionary must also be stored, each with their grammatical properties. Just from that, any user who conforms to the rules set forth in that grammar book and the vocabulary set forth in the dictionary can interact with the resulting NLP. But not of course not every one conforms to a grammar book.

Processing English is tricky due to all the special cases that can arise, but is possible. For the sake of simplicity we consider a Latin language example here. In Latin, sentence structure is encoded in the words them selves, not their arrangement in the sentence. A single root for a verb can have many different endings applied to it, each giving it a different meaning and forming a new word which is called conjugation. In English verbs are conjugated by positioning various forms of pronouns in front of participles or root of the verb. In English we still use some endings such as ‘s’ and ‘ing’. The latter ending turns a verb root into an imperfect participle. The former changes a noun from singular to plural. Here is an example sentence (Latin, then some equivalent English translations):

(L1) Puella ludum ambulo.

(E1) The girl walks to school.

(E2) The girl walks to a/the school.

(E3) The girl walks to her school.

(E4) The girl is walking to school.

(E6) The girl is walking to a/the school.

(E7) The girl is walking to her school.

E1 to E7 show slight variations on the English translations which might be eliminated by the context of a particular Latin sentence. The subject in E1 through E7 is ‘the girl’ and the verb is ‘walk’ (‘walking’

or ‘walks’). The object of the preposition ‘to’ is ‘school’. Notice that English offers various ways to qualify ‘school’ using indirect articles and personal possessive pronouns. In L1 the subject is ‘Puella’ which is said to be in the nominative case due to its ending. The prepositional object is ‘ludo’ which is said to be in the accusative which indicates “motion toward”, also due to its ending. It translates to ‘to school’. If the sentence had a direct object it would be in the accusative case also [7]. A very simple view of Latin at this point would dictate that we map the Latin statement to a truth function where its nominative case noun is the first argument and its accusative case noun is the second argument. It doesn’t matter where the words appear in the sentence, they still get mapped to the truth function the same way.

The lack of definite and indefinite articles (i.e. ‘the’ and ‘a’) also simplifies the parsing process. If the objects and subject of the English sentences are swapped in position the meaning of the statement is changed. There are still some complications even in Latin. The word ‘to’ is encoded into the word ‘ludum’ in L1 because it is in the accusative case but in general the accusative case doesn’t always indicate motion, it sometimes just indicates the direct object of the verb. Also, prepositions as their name may imply, must come before the words they modify and the same is true for adverbs, so the parser must watch for those where order and placement does matter [7]. In some cases if these placement rules are not followed, the meaning may still be clear to native readers but may not be clear at all to an automated parser and it is a daunting task to try to account for every exception. Dealing with the exceptions would be a job for an AI.

For the computer to make sense of the Latin sentence, it would first break the sentence into words, normally delimited by spaces, then each word would be scanned by it’s endings to determine its part of speech while watching out for word exceptions such as prepositions and adverbs. If an adverb is found, a verb would be looked for directly after it and if a preposition is found a noun would be expected next. By looking at the endings alone the program will find the sentence’s subject, verb(s) and object(s) (if they exist). The adverbs do actually have endings which indicate that they are adverbs unlike prepositions. Adjectives can be scanned based on their endings also, which bind them to certain nouns. To make the system complete would still require significant work in dealing with irregular verbs and nouns and other odd types of words that arise such as deponent verbs.

The information that is collected by the program is next stored in a data structure, along with other the necessary information about the words which was looked up in the dictionary (database) so they can be semantically analyzed later [11]. At the end of the parsing of a sentence, the program would expect to have the a minimal set of necessary components to form a grammatical clause. If the program

does not find all these components we should not assume the input sentence is ill formed as the input is from a natural language and may be slightly more complex than anticipated (i.e. “back to the drawing board”). But if the input can not be validated then no meaning can be found.

If all goes well the set of sentence components normally contains a subject and a verb at least. In Latin any conjugated verb is a valid clause because the noun is encoded into the word. The basic pronouns in English, which are encoded into Latin verbs are ‘I’, ‘we’, ‘you’, ‘you all’, ‘he’/’she’/’it’ and ‘they’. The list shows the singular and plural forms the 1st, 2nd and 3rd person pronouns [7].

In general the parsing program would move through a whole text sentence by sentence in this way, dedicating a data structure for each sentence. Upon completion of the parse any errors in the grammar of the document can be noted and the program can ask for clarification if necessary. If there are no errors the last task in a multi sentence text is to bind pronouns if possible. The meaning of the statements may not be found if there are still errors and clarification is not possible.

Computing Meaning

To create a NLP that can compute the meaning of objective statements, we need two main layers, a main natural language layer which converts the natural language text into some abstract form and then an abstract database of concepts in which the concepts are defined in terms of each other as much as possible. Using the database, the meaning of the original input statement can be found in a purely logical way. In the database every concept would be defined in terms of a single conceptual model. This database will be a dictionary basically, just like those used for word definitions in natural languages.

In a natural language dictionary (or NLD) each definition of each word is a word, sentence or sequence of sentences. Each word in each definition has a definition. So the dictionary can be used to compute the meaning of any word in a sense by following all the definition paths of every word in the definition of any word. But some definitions are circular and others may lead to an interminable path. Neither of those cases is computable. For the case of an interminable sequence we could just pick an arbitrary stopping point and for a cycle we could just end where we began. But the fact that these cases exist in a natural dictionary indicates that natural language is at least partly incompatible with a functional theory of meaning. The problem is one of inclusion: a natural language will include all and any words without regard for the interconnected structure which is formed by these words.

There is a subset of words in modern natural language dictionaries which abstractly have objective deterministic meaning. For a word to have deterministic meaning the definition tree which starts from one of these words would eventually reach a stopping point on all its branches and the full

tree (or set of paths) would yield the meaning of the word. A tree is just a linked structure which has a single root node and no cycles (i.e. loops back to previously visited nodes). Words which have this property are generally mathematical or scientific in nature. Also, since words can have multiple definitions, some of the definitions for words in the NLD will have this property and some will not. If multiple meanings exist for the same word then these different meanings must exist in a different lexical scope.

For an NLD to be structured in this way, the wording of definitions would have to be extremely precise. Some NLD words definitely have the potential to be made precise and those are the ones to focus on here. Some words of interest might be extremely technical in nature and appear only in technical dictionaries, but for the sake of simplicity let us just assume that there is some NLD that has a definition for every possible word in every academic field. Some online dictionaries might actually make that claim. In reality we might have to look at several dictionaries.

We can think of the building of this database as a buildint of meta languages. We start by building into the database the minimal meta language needed to describe more complex concepts. This would start with logical and mathematical language components. Next define scientific terms starting with more fundemental areas, then defining more complex words in terms of previously defined words. Science defines all physical objects and systems by observations and experiments. Words that can not be resolved as physical, mathematical or logical may have to be excluded until they can be assimilated. It is possible they can be added later, but if not then the whole structure must be rebuilt.

If the structure of the definitions of a word forms an actual tree then the sequence of definitions leads to computable meaning. A sentence of n words defined in a structure such as this might be transformed into a new statement of k words, where $k > n$, $k < n$ or $k = n$ and where the words in the new statement may or may not be the same as in the original. For instance, the meaning of the symbol '5' in mathematics might be expanded further as 'the successor of 4', or it might not be expanded any further if '5' is taken as a fundamental symbol, but in either case it can be expressed as a function of two other numbers such as '2+3'.

A Formal Language Dictionary

If an NLP is going to compute the meaning of a natural language statement, we may consider the NLP to be divided into two essential parts. The first part is initial parsing of the statement as described above. The second part is the mapping of the parsed data to logical structures in some pre-made database also talked about above. The following is intended to clarify the process further.

It is common practice for modern philosophers to translate their arguments from a natural language to a purely logical representation. This allows the argument to be checked within the framework of a formal language. For instance the statement 'The president of the United States is George Bush' can be translated into the logical statement: *president*(*u*) = *g* where *president*(*x*) is a function which represents the president of country *x* (assuming there is only one such position in country *x*), *g* represents 'George Bush' and *u* represents 'The United States'. Another example might be 'George Bush's father was the president of the United States' which could translate to @*time t - k*: *father*(*g*) = *president*(*u*) where *father*(*x*) represents the father of *x* , *t* is the current time and *k* is some other amount of time. The statement is saying that at some time in the past, the father of George Bush was the president of the US. These two statements regard public knowledge and so the value of those statements can be computed as true if we make certain assumptions about the subjects of the statement. [5]

The symbol @*time t:* is a modal operator which operates with Boolean valued functions. @*time t*: *P*(*x*) is true iff *x* has the property *P* at the time *t*. But we could just choose to make time part of any predicate if it were relevant to that predicate. For instance *P*(*x*,*t*) might mean that *x* has the property *P* at time *t* or that some event occurred involving *x* at time *t*. We could expand that to express places (i.e. spatial relations) also if we wanted. But for the sake of modularity, it makes more sense to let as many elements as possible be independent, so that the more complex elements can be defined by the combinations of the simpler ones.

The key concepts that should be captured in the formal language dictionary (or FLD) are the concepts that let us talk about the world as we know it: concepts made precise by science and mathematics. The precision of word definitions may vary from concept to concept though as some concepts are more fundamental and thus more important than others and some are based on different kinds of scientific information, such as biology vs. astronomy.

If we start with a natural language dictionary (NLD) we can organize its words into verbs, nouns, adverbs, adjectives and prepositions. Other types of words may not be necessary to consider here or they may just be defined in terms of these five types.

All the verbs can be expressed by either logical relations, properties or predicates. Each of those three logical entities is a truth function whose arguments are either logical objects or sets of logical objects. Nouns can be converted into logical objects by defining them in terms of iota notation or set builder notation. Iota notation just defines a singleton set, or in other words defines a unique object, in a class of its own. Adverbs and adjectives can be thought of as functions which take verbs or nouns respectively and return new verbs or new nouns. For instance, 'cat' is a noun and so is 'panther'. But a

'panther' is a 'black cat'. So we can replace 'panther' in a sentence with 'black cat' and the sentence is still true (if it was true to begin with). Since 'panther' is a noun, it's replacement is also a noun. The same logic holds for verbs: if a verb in a sentence can be replaced by a sequence of words then that sequence of words is a verb. This is not to say a substitution of this sort produces an equivalent sentence, it is strictly only true if the original sentence was true. Finally for prepositions, we may want to define them as functions which take verbs and return new verbs (or events). For instance, the preposition 'in' and the verb 'is' combine to form 'is in' which is a property or event, deepening on how you want to view it. Prepositions generally express special relations and relations of motion, as in motion towards or away. If something is in motion that motion is defined by a verb and must be bound to that verb. If something is specially related to something else then a claim is made about the a state of being of something. For instance: 'The cat is in the house.' The statement is saying that the cat is in a state of being in the house. The verb 'is' does some interesting things in English. For instance 'the cat is black' is really saying that the cat is in a state of being black which could father translate to a logical form as 'the cat has the property of being black.'

To create the FLD we first translate the most basic verbs into truth functions which take certain arguments. Some verbs take only one noun, a subject. Some take a subject and a direct object and some take subject, direct and indirect objects. Some verbs express physical events and in a sense serve as a basic model for the world as we write about it and some express more abstract concepts which allow further analysis and understanding of the world which we observe. When prepositions, adverbs and verbs bind together in a sentence predicate, a truth function with more than four arguments may be defined. Once a verb is converted into a truth function it will be define in terms of other truth functions. The result of this process is a definition tree for each word. The same is done for nouns: each noun is described as either being a unique thing or member of a class. Classes in turn are defined in terms of other classes. If the definition trees terminate then the words will have computable meaning. If not then the meaning is given by the deepest depth of the tree.

If the meaning of a statement involving all mathematical objects is sought, then the definition tree will terminate at either a number or a variable. If it terminates at a variable the result is an expanded algebraic expression. If the tree terminates at a number at all nodes then the result will be a number. This also works with statements that do not involve numbers as shown above with the example of the president of the US. In that particular example the best meaning that could be given might be to define what a president is, but the system would probably not have any information about particular people, so a true/false value could not be found.

Lets consider an example using natural statements regarding mathematics. 'The square root of four is less than four.' Clearly the meaning of this statement reduces simply to 'true' because the definition trees involved terminates. Here is how: the statement translates to '*sqrt*(4) < 4'. The symbol '2' is defined in the classes of numbers for which it is a member such as **N** and **R**. The *sqrt*(*x*) function is well defined and is a member of **R**, and for values like '4' is a member of the class of integers. So we can easily see that *sqrt*(4) = 2 and '2 < 4' is true. For a real number result, approximations would have to be computed.

Conclusion

A limited type of NLP could be created to work like a calculator for words and statements. Given a single word, it could give you the deepest meaning of that word and given a statement or sequence of statements it could either compute the value of the text as true or false or give its deepest meaning that the definition tree yields. This is best seen with examples using statements about math since the program Mathematica can already easily compute almost any mathematical statement value numerically or logically. It is just a matter of designing an NLP to translate certain natural language statements into Mathematica code. This NLP could ignore any words not relating to mathematics. The same approach will work for more general words regarding scientific concepts but requires an extensive database which would involves significant work to build. The main hurdle is that we would have to essentially translate a large subset of our natural language dictionary into propositional logic.

Reference Sources

#	Authors	Titles	Topics
1	Aaby, Anthony A	Logical Foundations of Computer Science and Mathematics	denotational semantics
2	Barrignton, David M.: UMass Department of Computer Science.	A Mathematical Foundation for Computer Science	computability, mathematical logic
3	Bartle, Robert G. Sherbert, Sherbert	Introduction to Real Analysis	high level math proofs
4	Cormen, Leiserson, Rivest, Stein.	Introduction to Algorithms	
5	Hardegree, Gary M.: UMass Department of Philosophy.	Symbolic Logic: A First Course Symbolic Logic: A Second Course Metalogic and Mathematical logic (unnamed book at http://people.umass.edu/gmhwww/513/text.htm)	natural and formal language semantics, formal languages, natural language translation
6	Hatcher, William S.	Logical Foundations of Mathematics	Zermelo Frankel, Freega, Russel
7	Oxford Latin	Oxford Latin	Latin, English grammar
8	Quine, W. V.	Methods of Logic	predicate logic, natural language translation
9	Serway, Raymond A. Jewett, John W.	Physics for Scientists and Engineers	
10	de Swart, Henriëtte	Introduction to Natural language Semantics	
11	Wikipedia	http://en.wikipedia.org denotational semantics, natural language processing, natural languages, semantic analyalisi	cross refereneing
12	Roget's Thesaurus	http://thesaurus.reference.com/	Six main classes of words