One difficult thing about natural language is that it is subjective and context dependant. For example, experience affects the use of language. Idioms are one such example of this, a person familiar with the use of an idiom will be able to make sense of it but a machine that only knows the technical meanings of words would not be able to sense the underlying message of an idiom. I.e. if one said “my car eats fuel” a machine may take this literally and assume the car consumes fuel whilst a human that has heard this phrase before would know that the car consumes a lot of fuel. This distinction is not something that is apparent from the sentence by itself and requires some experience with language in order to understand its meaning.

This raises the point that a machine needs to be able to experience language in order to understand it, but it cannot communicate until it has an understanding of language, so cannot get experience. Without being sure that it will be directed in its learning you cannot simply allow an artificially intelligent language processor to communicate with other entities and try to learn by itself via observation.

Morphology is the study of words and their relationships to other words. Determining what a sentence means when different words are combined, and how the grammatical and linguistic context of the words affects their meaning, is crucial in language processing. Words don’t have much meaning without context, it is through language that words form relationships with each other in order to form meaningful information about the world that they are describing. I.e. passing the words “had”, “cowboy”, “the” and “hat” are useless on their own and don’t mean much but “the cowboy had a hat” is a phrase that contains descriptive information about the world, since it assigns words to have relationships with other words (the hat being an object that belonged to the cowboy).

Parsing is performed on sentences passed into a natural language processor in order to evaluate what the sentences mean, and what parts of the sentence indicate what information. Parsing examines the grammatical structure of the language passed into the language processor and then exploits its knowledge of language rules in order to extract information from the words given to it.

Semantics is a field of knowledge that focuses on the meaning of words and how these words can be used to describe ideas, concepts and objects using other words which relate to ideas, concepts and objects, making it ultimately circular in nature and making all meaning relative. This is important to note in the field of artificial intelligence because an artificially intelligent agent cannot necessarily be said to have an underlying understanding of the subjective “meanings” of words, only possessing the connections between words that we provide. The distinction between an artificially intelligent system and human being seems intuitive but there is not necessarily a distinction as we assume that human beings have some sort of intuitive and fundamental understanding of words and ideas that is not simply circular and relative to other words and ideas but we cannot know if this is true for certain; since we do not fully understand the nature of knowledge.

The idea of semantics, which is usually a domain reserved for adaptive beings, being applied to a machine seems counter-intuitive but the principles can be said to be largely the same. Semantics create links between words and computers can form information structures, which are organised sets of information, such as lists of information, or pairs of words which are connected: this forms a sort of interconnected web of information. This web can be exploited to form pattern recognition and spot trends and links between information. The problem with this is that it is very broad and evolving whilst language is used to communicate ideas in a very specific and particular manner, which introduces the need for a more sophisticated approach to a use of language in order to accurately articulate the precise meaning of the information one is trying to communicate.

Our application is about an assassin that is trying to assassinate a target while manoeuvring a world of obstacles. The assassin is an agent in the world, the target is an agent that can also move autonomously and there are guards in the world that move autonomously and will try to stop the assassin if they cross paths with it. One point this raises is that communication is relative to the conversers and is not something said in a vacuum that applies to all language: the people in the conversation influence the context of the language. This aspect of language is difficult to artificially reproduce since it requires two subjective beings with potentially different ideas to communicate while a computer must be exact in the way it communicates. However, one thing higher level languages are capable of doing is using interfaces: this means they are capable of establishing contracts and agreed methods of doing things even if the underlying mechanism is encapsulated away from other parts of the language. This means if two different artificial entities support many of the same interfaces they will be able to communicate even if they are built differently, this is similar to two humans being able to communicate if they speak the same language with previously established rules and lexica.

The main idea we had for implementing language processing in our application was to have the end user be able to communicate with the assassin and give it commands. It is important to note that instead of making a statement there is a directed line of communication between the user and the assassin. This is an important distinction to make because context changes language; i.e. instead of saying “the assassin moves north” the user would say “move north” which would be understood as an instruction to the assassin which it would then attempt to carry out.

The idea of how we could implement natural language processing into this app would be to have a command line interface which would take natural language commands and convert them into information that the program can make use of. For example, if they give a command to the assassin it will evaluate the command and give the appropriate response. Either reject the command because it is either invalid or it does not understand it, perform an operation available to it (i.e. move north), or set a goal which it will attempt to work towards (i.e. kill the target).

Our program is primarily limited in the amount of operations it supports. Currently the assassin can only do three operations: move, pick up an object, or kill the target. This limits the use of language that the user is able to use, they cannot simply request anything of the assassin and have the assassin understand what they mean; it is limited by what is programmed in. Humans are able to avert this by being able to learn new words, building upon words and concepts that they already know (known via either previous learning through language or by observation of external stimuli).

The question of whether or not using natural language processing is appropriate for this program doesn’t have a definitive answer. On one hand, having a natural language interface may help some users to have intuitive command over the program but on the other hand, since it is a small number of possible operations, it may be a much better approach to simply use a series of buttons to perform actions.

This raises the question of when exactly is a natural language interface appropriate for use in a computer program. In most applications there is a specific domain in mind and a specific way of performing the operations it is programmed to do (i.e. in a messenger application there would be certain functions the application is capable of performing such as sending a message or adding a contact) which limits the language that the user can use and it may be easier for the user to just use more traditional cues in using the application.

Presumably the most appropriate place for a natural language interface would be an application that is very broad and where it would not be feasible to simply provide the user with buttons to press. One such example of this is an artificial personal assistant that is built into many modern operating systems. These act as language interpreters and are able to interpret a large number of commands, which acts as a shortcut to the user navigating through the operating system to perform those commands themselves. This is an appropriate use of natural language processing because it provides users the freedom of natural language that they would not have in an application with minimal commands.