

An Introduction to Natural Language Processing (NLP)

Peter Coxhead

Definition

A ‘natural language’ (NL) is any of the languages naturally used by humans, i.e. not an artificial or man-made language such as a programming language. ‘Natural language processing’ (NLP) is a convenient description for all attempts to use computers to process natural language.¹ NLP includes:

- Speech synthesis: although this may not at first sight appear very ‘intelligent’, the synthesis of natural-sounding speech is technically complex and almost certainly requires some ‘understanding’ of what is being spoken to ensure, for example, correct intonation.
- Speech recognition: basically the reduction of continuous sound waves to discrete words.
- Natural language understanding: here treated as moving from isolated words (either written or determined via speech recognition) to ‘meaning’. This may involve complete model systems or ‘front-ends’, driving other programs by NL commands.
- Natural language generation: generating appropriate NL responses to unpredictable inputs.
- Machine translation (MT): translating one NL into another.

Origins

- The idea of using digital computers in NLP is ‘old’, possibly because one of the first uses of computers was in breaking military codes in the second world war. Some computer scientists seem to have thought that Russian (for example) is just English in a different code. In which case, since codes can be broken, so can Russian. This idea assumes there is a common ‘meaning base’ to all natural languages, regardless of their surface differences. The overwhelming consensus among linguists is that this is simply not true.
- ‘Artificial Language Processing’, in the form of compilers and interpreters for programming languages, was a key component in the success of digital computers from their earliest days. This success undoubtedly encouraged research into NLP (and also encouraged an optimistic approach).

There have been cycles of optimism and pessimism in the field of NLP (we are possibly in a more optimistic phase at present); although some very real advances have been made, the target of a general NLP system remains elusive. Historically, computer scientists have often been far too over-optimistic about NLP, probably for some of the reasons noted above. It is thus important to be clear from the outset exactly why the task is difficult.

It is also important to note that there are differences between natural languages. More work has probably been done on English than on any other language, largely because of the importance of American researchers, although there are very active workers in Europe and Japan. However, English is in some ways an untypical language, as it uses few inflections and relies heavily on word order. Textbooks and other introductory sources written in

¹ NLP is often used in a way which excludes speech; SNLP is then needed as a term to include both speech and other aspects of natural language processing.

English rarely contain adequate discussions of NLP for languages with markedly different grammatical structures.

We can distinguish at least three distinct ‘levels’ in processing NL:

- Sounds
- Grammar
- Meaning

Each can be divided into two or more sublevels, which need not concern us here. What I want to do in this brief introduction is to illustrate some of the problems in processing each level.

Speech

Consider these three words, spoken by a native English speaker from the south of England: *input*, *intake*, *income*. It’s clear that all three words contain the element *in* with the same meaning. To input is to put something in; the intake of a water pump is the place where water is taken in; your income is the money that you earn, i.e. that comes in.

Is the element *in* pronounced the same in all three words (by the specified speaker)? Careful listening will show that it is not. The word *input* is pronounced as if spelt *imput*, whereas *intake* is pronounced as spelt. If we let η stand for the sound usually spelt *ng* in English (e.g. in words like *sing* or *singer*), then *income* is pronounced *in η come*.

I specified native English speakers from the south of England because many speakers of Scots English do NOT behave in this way; instead they consistently pronounce the first element of all three words as it is spelt, i.e. as *in* (as may all English speakers when speaking slowly and emphatically).

Interestingly, English speakers are generally quite unaware of these differences, both in their own speech and the speech of others. This is not because they cannot distinguish between the three sounds *m*, *n* and η . The three words *rum*, *run* and *rung* differ ONLY in these three sounds and are quite distinct to all native English speakers.

Another example of the same kind of phenomenon occurs in the plurals of English nouns. Consider the words *cat*, *cats*, *dog* and *dogs*. A native English speaker explaining how plurals are formed is likely to say something like “you add an *s* sound to the end.” Careful listening will show that *cats* is indeed pronounced with a *s* sound, but *dogs* is not: it ends with a *z* sound. Yet as with my previous example, English speakers don’t normally notice this difference. Again it isn’t because they can’t distinguish between *s* and *z* since *Sue* and *zoo* or *hiss* and *his* differ only in their *s* and *z* sounds.

The conclusion is that the sounds that native speakers ‘hear’ are not the sounds they make. This is important in both speech synthesis and speech recognition. When generating speech from written text, a synthesizer must not turn every *n* or *s* into an *n* or *s* sound, but must use more complex rules which mimic the native speaker. Similarly a speech recognition system must recognize the sounds in *rum* and *run* as distinct, but must NOT decide that *imput* and *input* are different words.

Even more awkward is the fact that this behaviour operates across word boundaries. Try saying these sentences quickly and in an ‘informal’ style:

When playing football, watch the referee.

When talking about other people, watch who’s listening.

When catching a hard ball, wear gloves.

You’ll find (at least if you speak the same dialect of English as me) that you say *whem*, *when* and *wher* respectively. This means that

- a speech synthesis system will not sound right if it simply uses pre-recorded words, since how these are pronounced changes depending on neighbouring words
- a speech recognition system must treat the three pronunciations of *when* as the same while distinguishing between the same sounds in *rum*, *run* and *rung*.

Even this example doesn't capture all the complexities. A native speaker of American English pronounces the *t* and *d* in the words *write*, *writer*, *ride*, *rider* differently from a native speaker of English English. If we ignore the very slightly different pronunciation of the *d*, the words will be pronounced roughly as *write*, *wrider*, *ride* and *rider*. That is, *write* and *ride* are clearly distinguishable whereas *writer* and *rider* are pronounced EXACTLY the same. Yet a native speaker of this dialect doesn't 'hear' this. The actual sounds used in the sentence *I'm a writer and I write books* will be something like *I'm a rider and I write books* but will cause no confusion whatsoever; a listener who is used to this dialect will hear *writer* not *rider*.

Conclusion: TO RECOGNIZE WORDS CORRECTLY REQUIRES SIMULTANEOUS PROCESSING OF SOUND, GRAMMAR AND MEANING.

Grammar

Grammar in this context refers to both the structure of words (morphology) and the structure of sentences (syntax). I'm only going to consider some syntax examples here.

Compilers process the syntax of a programming language without needing to understand it. Can we write similar programs to process the syntax of a NL without processing the semantics, i.e. without trying to understand what it means?

Consider:

*'Twas brillig, and the slithy toves
Did gyre and gimbal in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.*

Without knowing what Lewis Carroll² meant by some of these words, i.e. without being able to perform semantic processing, considerable syntactic processing is possible. Thus you can answer questions such as:

*What were the toves doing? – They were gyring and gimballing.
Where were they doing it? – In the wabe.*

(Notice that the answers may involve morphological changes to words in the original, e.g. *gyre* becomes *gyring*.)

Chomsky's famous

Colourless green ideas sleep furiously.

is another example of a syntactically correct sentence with erroneous or incomplete semantics. Again we can answer questions:

What were the ideas doing? – They were sleeping furiously.

Since people can answer such questions without needing to understand them, it seems at first sight plausible that a program could be written to do the same.

However, consider the second sentence of the fifth stanza of Carroll's poem:

*One, two! And through and through
The vorpal blade went snicker-snack!*

² Lewis Carroll was the pen-name of the mathematician and logician Charles L Dodgson (1832-98).

Syntactically the sentence is ambiguous. It could be equivalent to

Snicker-snack went through and through the vorpal blade.

or

The vorpal blade went through and through (something) snicker-snack.

depending on whether *snicker-snack* is taken to be a (plural?) noun or an adverb. (Consider *And through and through the wabe went borogroves* versus *And through and through the wabe went mimsily*.) However, there is actually no doubt as to the correct reading – things don't go through blades, and *snicker-snack* is onomatopoeic enough to be interpreted: the vorpal blade went through and through the Jabberwock making a snicker-snack noise.

More serious examples make the same point: in NLPs, syntax CANNOT be processed independently from semantics. Consider this sequence:

*The girl eats the apple with a smile.
How does the girl eat the apple? – With a smile*

Suppose we tried to write a computer program which could engage in conversations like this. The simplest approach seems to be to use pattern-matching. We might look for a sentence with the pattern:

The <noun1> <verb>s the <noun2> with a <noun3>.

Then if we had a question about this sentence which had the pattern:

How does the <noun1> <verb> the <noun2>?

the program could answer:

With a <noun3>.

Thus given:

The man closes the door with a bang.

the program could cope with the sequence:

How does the man close the door? – With a bang.

But now suppose we try the sentence:

*The girl eats the apple with a bruise.
How does the girl eat the apple? – With a bruise. WRONG!!*

The problem is that although on the surface the two sentences *the girl eats the apple with a smile* and *the girl eats the apple with a bruise* are similar, their underlying syntax is quite different: *with a smile* qualifies *eats* in the first sentence whereas *with a bruise* qualifies *the apple* in the second sentence. We can try to show this by using parentheses:

*[The girl] [eats (the apple) (with a smile)]
[The girl] [eats (the apple (with a bruise))]*

Unfortunately, to work this out our program will need to know that, for example, apples can have bruises but not smiles.

Now consider:

The girl eats the hamburger with relish.

This is ambiguous: it may mean that the hamburger has relish on it or it may mean that the girl is eating the hamburger with enthusiasm. Programming languages are deliberately constructed so that their syntax is unambiguous, but NLPs are not. Thus PROCESSING THE GRAMMAR OF NATURAL LANGUAGES DEPENDS ON SIMULTANEOUS PROCESSING OF MEANING.

Meaning

Meaning can be subdivided in many ways. A simple division is between semantics, referring to the meaning of words and thus the meaning of sentences formed by those words³ and pragmatics, referring to the meanings intended by the originator of the words. Note the meaning of sentences depends not only on the meaning of the words, but also on rules about the meaning of word combinations, word orders, etc. For example, in the English sentence *the boy sees the girl*, we know that the boy does the seeing and the girl is seen because in English the normal word order is subject – verb – object.

A striking feature of NLPs is that many words and sentences have more than one meaning (i.e. are semantically ambiguous), and which meaning is correct depends on the context. This problem arises at several levels.

There are problems at the level of individual words. Consider:

The man went over to the bank.

What kind of ‘bank’? A river bank or a source of money? Here we have two distinct English words with the same spelling/pronunciation.

However, there are more subtle problems at the word level. Consider:

Mary loves Bill.

Mary loves chips.

The word *loves* in these two sentences does not have precisely the same meaning (and might need translating by different words in other languages). This is especially clear if you try changing *Bill* to *chips* in:

Mary loved Bill – that’s why she killed herself.

Metaphorical use of words also causes problems. Consider:

Water runs down the hill.

The river runs down the hill.

The road runs down the hill.

In the first sentence, *runs* implies movement; in the third it does not. What about the second? Other languages may not allow the same word to be used in all three senses. The literal/metaphorical boundary tends to shift with time and usage.

There are also problems at the sentence level. The meaning of a sentence is not just the meaning of the words of course, it also involves knowledge of the rules governing the meaning of word combinations, orderings, etc. in the NL. However, even given this knowledge, idioms cause problems:

He really put his foot in it that time.

The classic (but I suspect fictional) example from MT is the system that translated:

The spirit is willing but the flesh is weak.

into Russian, and then translated the Russian back into:

The vodka is good but the meat is rotten.

At the passage level endless problems arise. ‘Anaphora’ (e.g. substituting less meaningful words such as pronouns for nouns) is a common feature of NLPs. Suppose we read the sentence:

London had snow yesterday.

³ In some languages, some words may have a purely syntactic function and should be excluded from this analysis; e.g. *to* in *I want to eat*.

and then read ONE of the following sentences, all of which are sensible continuations. What is the *it*?

It also had fog. (It = London)

It fell to a depth of 1 metre. (It = the snow)

It will continue cold today. (It = ?the weather)

Handling pronouns is a difficult task, and seems to require considerable world knowledge and inference. (Note that even in MT where ‘understanding’ the text is apparently not an issue, inferring the original of pronouns can be important, because it may be necessary to reflect gender; e.g. if translating the above into French, *it* may become either *il* or *elle*.)

Conclusion: THE MEANING OF NATURAL LANGUAGE CANNOT BE ESTABLISHED FROM THE MEANING OF THE WORDS PLUS RULES FOR DETERMINING MEANING FROM WORD COMBINATIONS AND ORDERINGS; INFERENCE AND WORLD KNOWLEDGE ARE NEEDED.

If all this weren’t enough, we need to consider the (inferred) intention of the producer of the NL being analysed. It may be thought that this is covered by semantics, but a few examples soon show otherwise. Consider the question:

Can you tell me the time?

A syntactically and semantically perfectly correct response is *Yes!* Pragmatically, this is wrong (assuming the question was asked in a ‘normal’ context by a native speaker of English), since the intention behind the question is that the respondent should TELL the questioner the time. How can a program be written which deals correctly with the difference between

Can you swim?

and

Can you pass me the life belt?

The difference between

Pass me the salt.

Pass me the salt, please.

Can you pass me the salt?

Could you pass me the salt?

is essentially in levels of politeness, rather than in any other aspect. This issue is particularly important in MT, where literal (i.e. syntactically and semantically correct) translations may not be pragmatically so. For example, these Greek-English pairs are reasonably good translations as regards syntax and word/sentence semantics:

thélo mía bíra = I want a beer

tha íthela mía bíra = I would like a beer

but the first Greek sentence is more polite than the first English sentence, so that a better translation may be the second English sentence.

Conclusion: THE SURFACE MEANING OF A SENTENCE IS NOT NECESSARILY THE MEANING INTENDED BY THE PRODUCER.

Conclusion

NLs are MASSIVELY LOCALLY AMBIGUOUS at every level (speech, grammar, meaning). Yet in normal use, NL is effective and rarely ambiguous. To resolve local ambiguity, humans employ not only a detailed knowledge of the language itself – its sounds, rules about sound combinations, its grammar and lexicon together with word meanings and meanings derived from word combinations and orderings – but also:

- A large and detailed knowledge of the world (e.g. knowing that apples can have bruises but not smiles, or that snow falls but London does not).
- The ability to follow a ‘story’, by connecting up sentences to form a continuous whole, inferring missing parts.
- The ability to infer what a speaker meant, even if he/she did not actually say it.

It is these factors that make NLs so difficult to process by computer – but therefore so fascinating to study.