Hyperiondev

TASK

Introduction to Natural Language Processing

Visit our website

Introduction

WELCOME TO THE NATURAL LANGUAGE PROCESSING TASK!

This is an introduction to Natural Language Processing, which is one of the biggest current fields of research in Artificial Intelligence.

First, let's get you started with some background knowledge. By the end of this task, you'll be able to build your own AI program that can automatically classify tweets using the same technology that Google uses to filter spam from your email inbox!

Get in touch

Connect for support

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to **www.hyperiondev.com/portal** to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

AN INTRODUCTION (USING IRON MAN)

The image above is from the movie Iron Man. In this movie, the fictional main character, Tony Stark, is a billionaire genius engineer. He creates a suit of armour that is so powerful and advanced that he becomes a superhero nicknamed 'Iron Man'.

Tony has a robot AI inside his mansion called Jarvis. He talks to this robot many times during the film and it helps him perform certain tasks. The picture above is from a scene where Tony has just spoken to Jarvis.

Let's think about this a bit. What is the most advanced technology or a crazy idea in the movie Iron Man?

- Is it the fact that Tony has built a suit made out of metal that he can fly around in?
- Is it the fact that the suit makes Tony so strong that he can shoot missiles from it, reflect bullets and fly around?
- Is it the fact that Tony is a billionaire engineer that is smart enough to do this by himself?
- Or is it the seemingly small and insignificant fact that Tony can talk to the Artificial Intelligence 'Jarvis' and Jarvis can understand exactly what he says?

If you didn't know better and had no background in Artificial Intelligence, you may think that flying around in a suit shooting missiles is more advanced than a robot understanding the few simple things that Tony says to it. But you'd be wrong!

The fact that Jarvis understands Tony's simple words "Wake up, daddy's home" and can reply correctly with "Welcome home, sir" is a massive technological feat of artificial intelligence, and problems of creating superhero suits and flying around in one is actually nothing compared to the huge field of Natural Language Processing, which is the main area of research in the field of Artificial Intelligence today.

Even before Ironman, there have been movies about space travel with people travelling on spaceships that have AI that can understand the crew speaking and reply to them. Today we have travelled space, gone to the moon, yet we have still failed to produce artificial intelligence systems that can do what is shown in these movies. How can this be the case? How can Natural Language Processing be harder than going to the moon?! How can it still be a totally unsolved area in Artificial Intelligence?

How would we even start?

NATURAL LANGUAGE PROCESSING

In order to start thinking about creating Jarvis in real life (i.e. a robot that we can talk to, understand what we say, and act on it or even just reply correctly) we'd need many things.

We call programs like Jarvis that converse with humans in natural language 'conversational agents' or 'dialogue systems'. Natural languages are languages humans talk to each other in; 'formal languages' are programming languages like Python or Java.

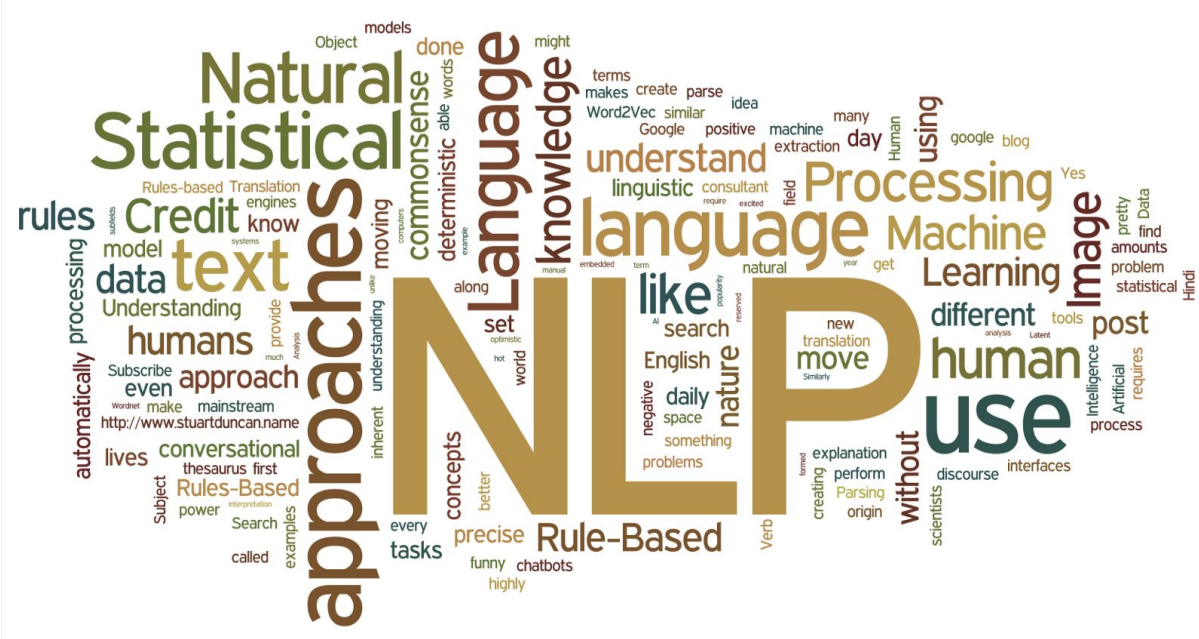
Jarvis must be able to recognise words from an audio signal and to generate an audio signal from a sequence of words. These tasks of speech recognition and speech synthesis require knowledge about phonetics and phonology: how words are pronounced in terms of sequences of sounds and how each of these sounds is created acoustically. Pronouncing variations of words correctly (such as plurals, contractions) requires knowledge about morphology – the way words break down into component parts that carry meanings.

What if we asked Jarvis, "How many University of KwaZulu-Natal university students are in the Math130 class by the end of the day?" Jarvis needs to know something about lexical semantics – the meaning of all the words (e.g. 'class' or 'students') and compositional semantics (what exactly makes a student a 'University of KwaZulu-Natal student' and not another type of student?). What

does ‘end’ mean when combined with ‘the day’? Jarvis needs to know about the relationship of the words to each other – how does Jarvis know that ‘by the end of the day’ refers to a time and doesn’t refer to something like ‘the book that is written by the author JK Rowling’? Humans know this automatically, but how can computers learn this?

How does Jarvis know that when Tony says ‘Daddy’s home’, Tony is actually talking about himself? Jarvis knows this because he says ‘Welcome home, sir’, so clearly he understood that somehow. This knowledge about the kind of actions that speakers intend by their use of sentences is pragmatic or dialogue knowledge. To summarise, Jarvis needs the following knowledge of language:

- Phonetics and Phonology – knowledge about linguistic sounds
- Morphology – knowledge of the meaningful components of words
- Syntax – knowledge of the structural relationships between words
- Semantics – knowledge of meaningful
- Pragmatics – knowledge of the relationship of meaning to the goals and intentions of the speaker
- Discourse – knowledge about linguistic units larger than a single utterance



Ambiguity

Now, what if Tony was telling Jarvis a story about his female assistant who had annoyed him? As it happened, Tony threw a piece of paper at her and she ducked to avoid it. Describing the incident, Tony says the following sentence to Jarvis:

“I made her duck”.

This simple sentence has the following meanings. The correct meaning in Tony’s story is in bold:

- I cooked waterfowl for her
- I cooked waterfowl belonging to her
- I created the (plaster?) duck she owns
- **I caused her to quickly lower her head or body (to avoid something)**
- I waved my magic wand and turned her into a common waterfowl.

But how can we make Jarvis smart enough to know this? There are many ambiguities in this sentence because just the word ‘duck’ can be a verb (move your head down) or a noun (a waterfowl), ‘her’ can mean the women or can refer to the fact that the duck belongs to her.

What about hearing? Say the word ‘I’ out loud. How does Jarvis know that this word isn’t actually ‘eye’. What about ‘made’? It sounds just like ‘maid’! Poor Jarvis!
We must use complicated models and algorithms as ways to resolve or disambiguate (remove ambiguities) these ambiguities.

Part of speech tagging

Deciding whether ‘duck’ is a verb or a noun is known as part of speech tagging. Verbs and nouns are each different ‘parts of speech’ and we ‘tag’ a word in a sentence by assigning it one part of speech that we think is correct for the context or sentence it has been used in.

In a phrase like “The old man and the boat”, the task of tagging the correct parts of speech may involve the following:

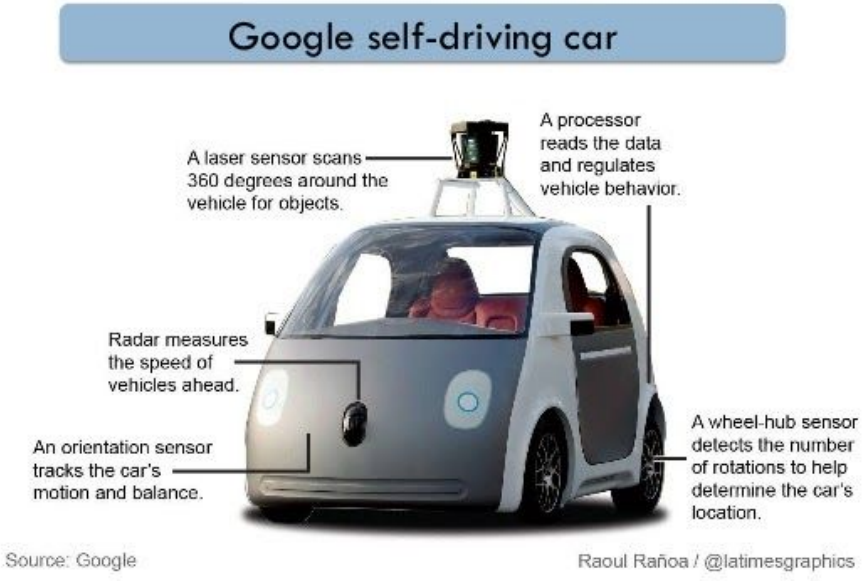
The: tag as ‘determiner’
old: tag as ‘adjective’
man: tag as ‘noun’ (or verb?)’
the: tag as ‘determiner’
boat: tag as ‘noun’

Sometimes, probabilities are used to decide this. For example, the word ‘man’ above is more probable to mean the noun ‘an older male person’ than the verb ‘put someone there’ (e.g. “The enemy is here! Man the cannons!”).

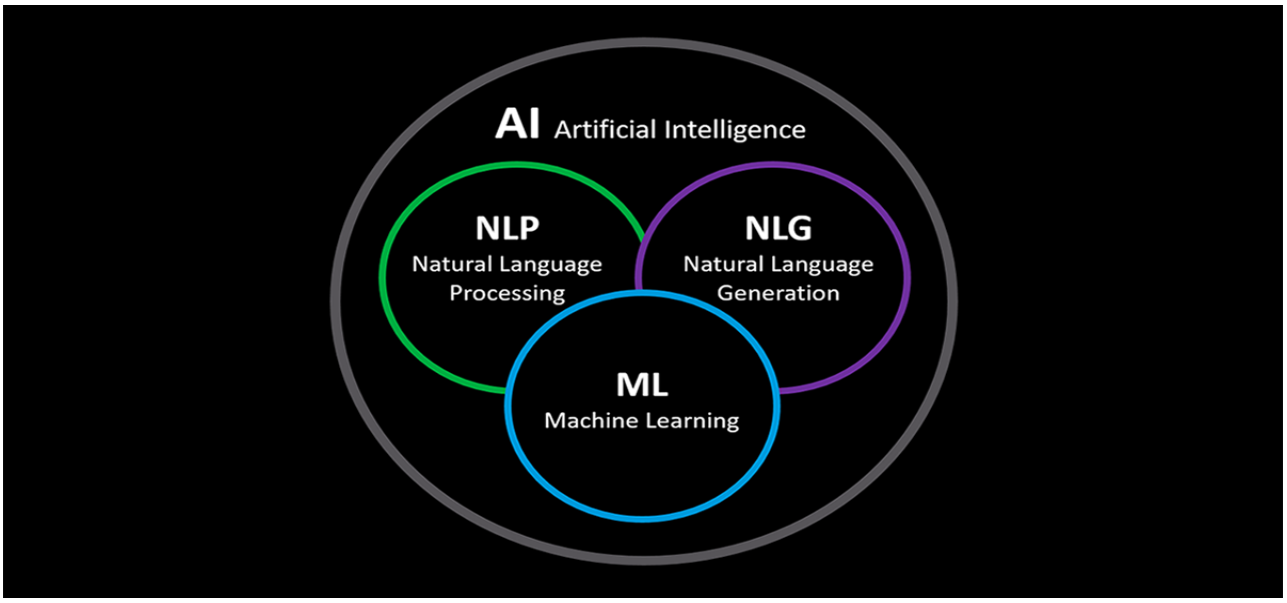
As you can see, Natural Language Processing (NLP) is a huge field and extremely important in the field of Artificial Intelligence.

NLP also includes the study in things like Machine Translation which is the same as the technology behind Google Translate – automatically translating between two languages. Google Search uses NLP techniques in order to understand your searches faster and this is what makes Google Search more accurate, reliable and faster than any other search method on the internet. NLP is used in Gmail in order to identify spam mail and delete it.

Natural Language Processing research is one of the main reasons Google is so successful. The probabilistic, machine learning techniques that Google applies to NLP can be applied to other Artificial Intelligence tasks such as creating driverless cars (currently in operation in California).



We hope you can see how many fields in Artificial Intelligence have to do with probability. This is because it is the only way we can deal with ambiguity to try to make robots/AI programs to make the best decisions!



MACHINE LEARNING

Natural Language Processing is a big part of Artificial Intelligence because a large amount of it has to do with training computers or AI programs to identify certain patterns and use probabilities to make informed decisions.

This is known as Machine Learning and is a massive field of research currently. Facebook uses Machine Learning to try to recommend friends to you, Amazon uses it to recommend items to buy, Google Image search uses machine learning techniques to identify patterns in pixels to try to find similar images, Google has designed driverless cars to act according to their environment by integrating many different machine learning. The list is endless.

POS Tagging

One example is to solve the problem of part of speech (POS) tagging as explained earlier. We can give a program a big set of tagged words (training data) and then give it a new Sentence whereby it must try to tag with the information learnt from the training data.

The University of Pennsylvania had the first NLP research program to ever take a large corpus (body) of words and tag each and every one by hand see: (**http://en.wikipedia.org/wiki/Treebank**). A program was then activated and slowly learnt how to tag words and the probability that a certain word appeared

with a certain tag in a certain context. Ever since then, AI runs on huge sets of training data to be more accurate.

In POS tagging we try to tag words with the correct part of speech tag so that we can then parse the sentence correctly. **Parsing** is the formal term for 'putting a sentence together in the right way' so that it can be 'understood' . We will talk about this later.

Text classification

We also use Machine learning to try to classify a text. For example, Gmail classifies emails as either 'spam' or 'not spam'. It uses machine learning by having trained an AI program on a set of already 'classified' emails (examples of non-spam and spam emails). Then when the AI sees a new incoming email, it can use its prior knowledge or 'training' to classify the new email correctly.

This task will have an example of how we can use Machine Learning to get a program to identify positive or negative tweets – similar to the problem of identifying spam mail through machine learning but first, we need to start with SpaCy.

STARTING WITH SPACY

SpaCy is a Python natural language processing library specifically designed with the goal of being a useful library for implementing production-ready systems (documentation **here**). It is particularly fast and intuitive, making it a top contender for beginners in NLP. Before doing anything, you need to have spaCy installed, as well as its English language model.

Type the following commands in the command line to install spaCy:

```
pip3 install spacy
```

To confirm that it is well installed, get into the Python console by typing the following:

```
# get into python console
python3

import spacy
```

If you receive no error, this means that spaCy was installed correctly!

If you face an error, please contact your mentor immediately. To exit the Python console, use the following:

```
quit()
```

Now let's talk about the language models.

spaCy is not very useful without at least one language model. The model allows you to process different languages. SpaCy so far has 8 language models which include French, Dutch, Spanish and of course English. You can install more than one model or even install a multi-language model all at once. SpaCy's models can be installed as Python packages. This means that they're a component of your application, just like any other module. Models can be installed from a download URL or a local directory, manually or via pip. Their data can be located anywhere on your file system.

To download the English model, type the command below in your terminal.

```
python -m spacy download en
```

To test our newest model, we will need to try using the model. Get into the python console and import spaCy as below

```
python3
import spacy
```

Now we will load the model and assign it to a variable.

```
nlp = spacy.load('en')
```

The input to natural language processing will be a simple stream of Unicode characters (typically UTF-8). Basic processing will be required to convert this character stream into a sequence of lexical items (words, phrases, and syntactic markers) which can then be used to better understand the content.

For spaCy, you can do this by passing the string through the language model you imported at the beginning of the script. Remember we named our model **`nlp`** so to process our string in preparation for spaCy manipulation, look at the code below:

```
doc = nlp("this is a test sentence")
print([(w.text, w.pos_) for w in doc])
```

Now that we are able to process a string we can do more complicated stuff. All the fun stuff is in the **example.py** file.

Instructions

In this task, we will use the spaCy, which is an external Python module that must be installed. Please contact your mentor ASAP if you can't get the 'import spacy' statement to work!

First, read **example.py** and run it. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this task to become more comfortable with the topic.

Compulsory Task 1

Follow these steps:

- This task will be to follow through the example file and install SpaCy.
- You'll be required to follow through the basics of spaCy and try to apply these concepts in a few sentences and give a short explanation.
- You can work through the **example.py** file to see further requirements.

Completed the task(s)?

Ask your mentor to review your work!

Review work

Acknowledgements:

This task uses text, code and slides adapted with permission from the University of Edinburgh's Informatics department, one of the leading Natural Language Processing research departments in the world. [The course content referred to is University of Edinburgh's Informatics 2A: Processing Formal and Natural Languages. All course content can be seen here: **http://www.inf.ed.ac.uk/teaching/courses/inf2a/**.]

Still need help? Just write your queries in your comments.txt file and your mentor will respond to your queries.

Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your mentor for alternative instructions.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.