LING 388: Computers and Language

Lecture 5
Today's Topics

• Text summarization
• Homework 3
• Python introduction contd.
Text Summarization

• Software:
  • available on Macs (turned off by default?)
  • Open Text Summarizer (ots) (on Linux)
Human Summarization

• Read this article now:

Let's summarize it!
Elon Musk recently commented on Twitter (TWTR, -1.67%) that artificial intelligence (AI) is *more dangerous than North Korea*. It’s not the first time that the entrepreneur has warned about the dangers of AI. Should we all be afraid as he is? Will AI lead to a huge disaster or robot takeover that destroys humanity?

Let’s be clear: A nuclear conflict between North Korea and the U.S. would cause millions to die. In the short term, AI is not going to lead to something worse than that. The AI technologies of today are simply not advanced enough or sufficiently embedded into our society for that to happen.

But in the long term, the outlook is less clear. AI technologies are developing fast and so are their attendant risks. AI applied to warfare and policing is certainly a concern. Autonomous armed robots, which can track and target people using facial recognition software, are just around the corner. Let loose, such machines would keep on killing until they ran out of targets or ammunition. This reminds us that AI has no social awareness, conscience, mercy, or remorse. It simply does what it’s been trained to do.
Elon Musk on AI

The "Terminator Scenario" is a real possibility, but it’s something that we can easily foresee and develop measures to protect ourselves against via legislation and in how we design robot intelligence. Arguably the greater threat from AI comes from developing machines that are better decision makers than we are. As a consequence, we could become the slaves of automated decision makers and whoever controls them.

This may seem like a futuristic nightmare, but it’s an insidious creeping process that is well underway. AI-based systems are beginning to replace or enhance many white-collar jobs that are currently done by human experts. This is because AI can interpret situations and make better calls than humans in those jobs. For example, AI machines are better at spotting the signs of cancer on a medical scan than the best radiologists are.

Many of our key life choices on social media and shopping sites are influenced by AI. Algorithms determine what content we see on our News Feed and recommend who we should date. People don’t set your insurance premiums anymore, and if you can’t get a loan, it’s because an AI-based system denied it. Many companies are using AI to screen job applicants for roles. In a few decades it may be that our lives are so controlled that we find ourselves unable to deviate from the life path that AI decides—we will be slaves to a machine-determined future.

What makes this scenario so dangerous is that it isn’t being planned by some overarching master intelligence or machine overlord. We are creating the very technology that could lead to our demise. This makes AI difficult to protect against or control. Instead, machines would take over in a piecemeal fashion. Whether it arrives today or in decades, the threat posed by AI is real.

Steven Finlay is the author of Artificial Intelligence and Machine Learning for Business: A No-Nonsense Guide to Data Driven Technologies, which was published in June 2017.
Open Text Summarizer (ots)

(code: read about how it works)

• https://github.com/neopunisher/Open-Text-Summarizer/
• Web interface
  • http://www.splitbrain.org/services/ots
Apple's Text Summarizer

- System Preferences (10.12 or 10.13)
Apple's Text Summarizer

• Let’s try it on the Elon Musk article:

Apple's Text Summarizer

1. Highlight text
2. Control-click for menu...
Apple's Text Summarizer

We Should Be as Scared of Artificial Intelligence as Elon Musk Is
Homework 3

• due next Wednesday by midnight to TA Patricia Lee

Questions:

1. Read the following article.
   Run the OTS on it. What does it report for 5% summarization?
   (You should copy and paste the text in rather than use the URL.)
   Did it do a good job summarizing the article or not? Explain.

2. Write one or two sentences to summarize the article.

3. Read the How It Works (scroll down!) in https://github.com/neopunisher/Open-Text-Summarizer/
   Then explain whether it’s plausible that a computer program could come with your sentence(s), or something similar.
Homework 3

• **Clickbait:**
  
  “Clickbait headlines typically aim to exploit the "curiosity gap", providing just enough information to make readers curious, but not enough to satisfy their curiosity without clicking through to the linked content.” (Wikipedia)
Homework 3

Find a clickbait article on the internet
(give the URL, copy text of article into Homework submission if short)
5. Summarize the article in one or two sentences
6. Use the OTS on the text. Show the 5% summary
7. In your opinion, did the OTS summary bridge the **curiosity gap**?
Python

- [https://docs.python.org/3/tutorial/introduction.html](https://docs.python.org/3/tutorial/introduction.html)
- Numbers
- Strings
- Lists
- Dictionaries
3.1.3. Lists

Python knows a number of compound data types, used to group together other values. The most versatile is the list, which can be written as a list of comma-separated values (items) between square brackets. Lists might contain items of different types, but usually the items all have the same type.

```
>>> squares = [1, 4, 9, 16, 25]
>>> squares
[1, 4, 9, 16, 25]
```

Like strings (and all other built-in sequence type), lists can be indexed and sliced:

```
>>> squares[0]  # indexing returns the item
1
>>> squares[-1]
25
>>> squares[-3:]  # slicing returns a new list
[9, 16, 25]
```
Python Lists vs. Sets

```python
>>> list = ['apple', 'orange', 'pear']
>>> len(list)
3
>>> s = set(['apple', 'orange', 'pear'])
>>> s
{'orange', 'pear', 'apple'}
>>> 'orange' in s
True
>>> 'banana' in s
False
>>> 'banana' not in s
True
>>> 'banana' not in list
True
>>> 'orange' in list
True
>>> list = ['apple', 'orange', 'pear', 'apple']
>>> list
['apple', 'orange', 'pear', 'apple']
>>> s = set(['apple', 'orange', 'pear', 'apple'])
>>> s
{'orange', 'pear', 'apple'}
```

- `in` – membership
- `not in` – not a member of
- `set(List)` - produces a set, no duplicates permitted
- Set is faster than list for lookup (hashtable)
5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.append(x)`
Add an item to the end of the list. Equivalent to `a[len(a):] = [x].`

`list.extend(iterable)`
Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable.`

`list.insert(i, x)`
Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x).`

`list.remove(x)`
Remove the first item from the list whose value is `x`. It is an error if there is no such item.
Python Lists

• Lists as stacks
• Lists as queues
• List Comprehensions (*advanced topic*)

https://visualgo.net/en/list?slide=4

https://www.appcoda.com/ios-concurrency/
Python List as a Queue

**EXAMPLE:**

```python
>>> list = ['c1','c2','c3']
>>> list[0]
'c1'
>>> list = list[1:]
>>> list
['c2', 'c3']
>>> list.append('c4')
>>> list
['c2', 'c3', 'c4']
```

- Method `append()` to add to the queue
- `list[0]` gives us the head of the queue
- `list = list[1:]` deletes the head of the queue from the queue