Natural Language Understanding and Language Models

Kyle Richardson

Allen Institute for Artificial Intelligence (AI2)

December 2020

Natural Language Processing: Goals

- Can we build machines that **understand** natural language (e.g., **English**, Dutch, Russian, ...)? Can a machine **reason/think**?

Natural Language Processing: Goals

- Can we build machines that **understand** natural language (e.g., **English**, Dutch, Russian, ...)? Can a machine **reason/think?**

Natural Language Processing: Goals

- Can we build machines that **understand** natural language (e.g., **English**, Dutch, Russian, ...)? Can a machine **reason/think?**

  *"The question of **whether machines can think** is about as relevant as the question of whether **submarines can swim.**' **Edsger Dijkstra***

Natural Language Processing: Goals

- Can we build machines that ~~**understand**~~ natural language (e.g., **English**, Dutch, Russian, ...)? Can a machine ~~**reason/think**~~**?**

  Alan Turing's Suggestion: Rather than asking such questions, devise tests that demonstrate understanding/reasoning/thinking.

Natural Language Processing: Goals

- Can we build machines that ~~**understand**~~ natural language (e.g., **English**, Dutch, Russian, ...)? Can a machine ~~**reason/think**~~**?**

Semantic Text similarity: Given a sentence (**start**) + a list of sentences (**sentences**), find the sentence in the list most related to the first.

```python
## input sentence
start = "A bulldog is playing in a field."
### sentences to compare against
sentences = [
        "The dog is running",
        "A book is laying on a bookshelf.",
        "Mary is sitting in the cafe with John."
]
def most_similar(target,sentence_list):
    ## return most similar sentence to `target` in
    ## sentence_list
most_similar(start,sentences)
```

Natural Language Processing: Goals

Semantic Text Similarity: Given a sentence (**start**) + a list of sentences (**sentences**), find the sentence in the list most related to the first.

```
## input sentence
start = "A bulldog is playing in a field."
### sentence to compare against
sentences = [
        "The dog is running",
        "A cat is laying on a bookshelf.",
        "Mary is sitting in the cafe with John."
]
def most_similar(target,sentence_list):
    ## return most similar sentence to ‘target‘ in
    ## sentence_list
most_similar(start,sentences)
```

Research Topic: Can models mimic the behavior of humans?; **Real-world applications**: Document search, retrieval, web search, paraphrasing.

The First Step: what would a human do?

Semantic Text Similarity: Given a sentence (**start**) + a list of sentences (**sentences**), find the sentence in the list most related to the first.

```
## input sentence
start = "A bulldog is playing in a field."
### sentence to compare against
sentences = [
        "The dog is running",
        "A cat is laying on a bookshelf.",
        "Mary is sitting in the cafe with John."
]
def most_similar(target,sentence_list):
    ## return most similar sentence to ‘target‘ in
    ## sentence_list
most_similar(start,sentences)
```

Conditions: *A sentence is similar to another sentence if: It has similar words ; has a similar syntactic structure ; a related subject/object , ..*

The First Step: what would a human do?

Semantic Text Similarity: Given a sentence (**start**) + a list of sentences (**sentences**), find the sentence in the list most related to the first.

```
## input sentence
start = "A bulldog is playing in a field."
### sentence to compare against
sentences = [
        "The dog is running",
        "A cat is laying on a bookshelf.",
        "Mary is sitting in the cafe with John."
]
def most_similar(target,sentence_list):
    ## return most similar sentence to `target` in
    ## sentence_list
most_similar(start,sentences)
```

Conditions: *A sentence is similar to another sentence if: It has similar words; has a similar syntactic structure; a related subject/object, ..*

Observations: Rely on fuzzy notions of similarity (e.g., `dog` vs. `bulldog` vs. `cat` vs. `Mary`); Constraints are not absolute (**soft constraints**)

How to Operationalize Fuzzy Notions in the Computer?

- **Not a new idea:** Represent linguistic objects (e.g., words) as continuous vectors: $\vec{\textbf{bulldog}} = [1., 3.], \vec{\textbf{dog}} = [2., 2.], \vec{\textbf{cat}} = [6., 7.]$[1]

[1] For a nice overview, see *Geometry and Meaning* (2004), Dominic Widdows.

How to Operationalize Fuzzy Notions in the Computer?

- **Not a new idea:** Represent linguistic objects (e.g., words) as continuous vectors : $\vec{\textbf{bulldog}} = [1., 3.], \vec{\textbf{dog}} = [2., 2.], \vec{\textbf{cat}} = [6., 7.]$[1]

$$\textsc{similarity}(\vec{\textbf{dog}}, \vec{\textbf{bulldog}}) = \sqrt{(2-1)^2 + (2-3)^2} \approx 1.41$$

[1] For a nice overview, see *Geometry and Meaning* (2004), Dominic Widdows.

How to Operationalize Fuzzy Notions in the Computer?

- **Not a new idea:** Represent linguistic objects (e.g., words) as continuous vectors: $\vec{\textbf{bulldog}} = [1., 3.], \vec{\textbf{dog}} = [2., 2.], \vec{\textbf{cat}} = [6., 7.]$

$$\textsc{similarity}(\vec{\textbf{dog}}, \vec{\textbf{bulldog}}) = \sqrt{(2-1)^2 + (2-3)^2} \approx 1.41$$

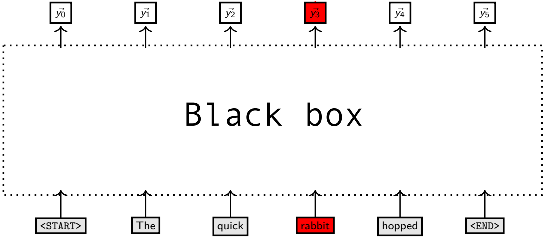
Recent work: generalize to more complex linguistic objects[2].

```
## input sentence
start = "A bulldog is playing in a field."
### sentence to compare against
sentences = [
        "The dog is running",
        "A cat is laying on a bookshelf.",
        "Mary is sitting in the cafe with John."
]
### vectors for sentences
print(sentence_representation(
             "A bulldog is playing in a field"))
array([ 2.50454932e-01, -1.26090437e-01,... ,
       -2.36674145e-01, -2.65906781e-01, ,...
       -2.34061420e-01, -4.75490779e-01, ....
        2.80727118e-01,  5.04251420e-01,....
        2.22773388e-01, -7.11838424e-01,....
       -1.19844747e+00, -7.89292753e-01,...,])
```

[2] Generated using *Sentence Transformers*, see more details at `https://www.sbert.net/`.

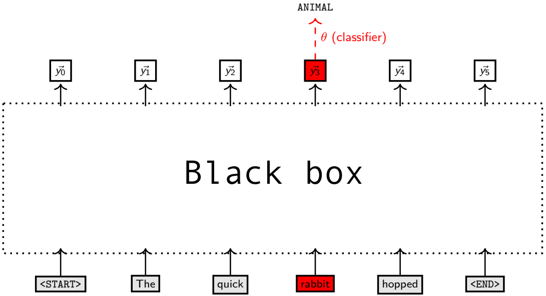
Language Models: Learning Contextual Representations

- **Role**: assign continuous vector representations $\vec{y} \in \mathbb{R}$ to inputs based on their meaning in each instance; deep neural networks.



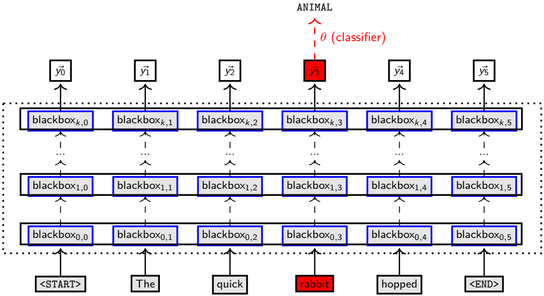
Language Models: Learning Contextual Representations

- **Role**: assign continuous vector representations $\vec{y} \in \mathbb{R}$ to inputs based on their meaning in each instance; deep neural networks.



Language Models: Learning Contextual Representations

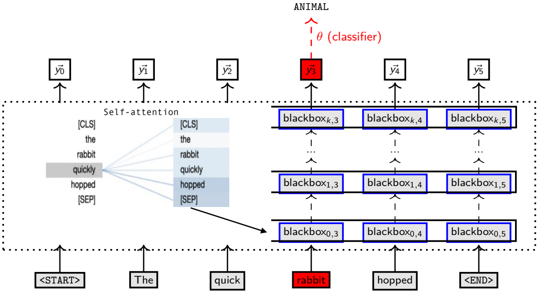
- **Role**: assign continuous vector representations $\vec{y} \in \mathbb{R}$ to inputs based on their meaning in each instance; deep neural networks.



- **Development 1**: New neural architecture called **Transformers**, internally involve a kind of brute force *looking around*.

Language Models: Learning Contextual Representations

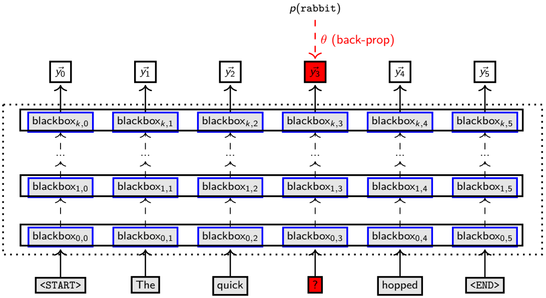
- **Role**: assign continuous vector representations $\vec{y} \in \mathbb{R}$ to inputs based on their meaning in each instance; deep neural networks.



- **Development 1**: New neural architecture called **Transformers**, internally involve a kind of brute force *looking around*.

Language Models: Learning Contextual Representations

- **Role**: assign continuous vector representations $\vec{y} \in \mathbb{R}$ to inputs based on their meaning in each instance; deep neural networks.



- **Development 2**: **Model pre-training**: Have the model read the internet (terabytes of data) and learn by solving word completion (*cloze*) tasks.

How do we find the right representations?

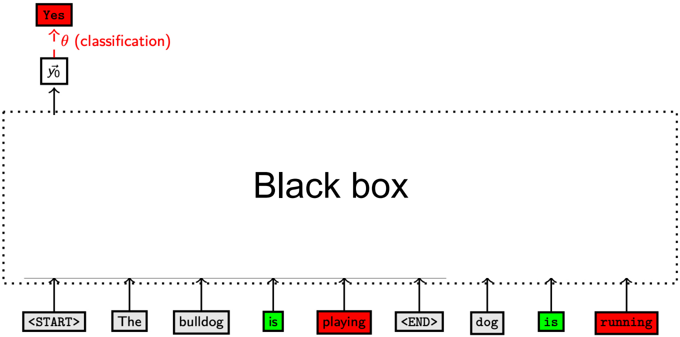
- This is the part that involves **machine learning**; optimization.

Machine Learning

From xkcd

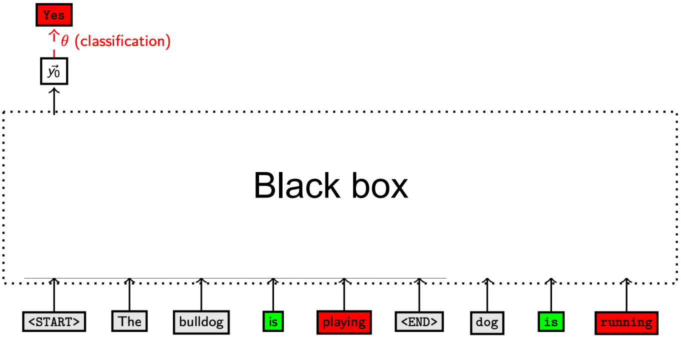
Other Uses of Language Models

- Building specialized models on top of basic model; **classification**.



Other Uses of Language Models

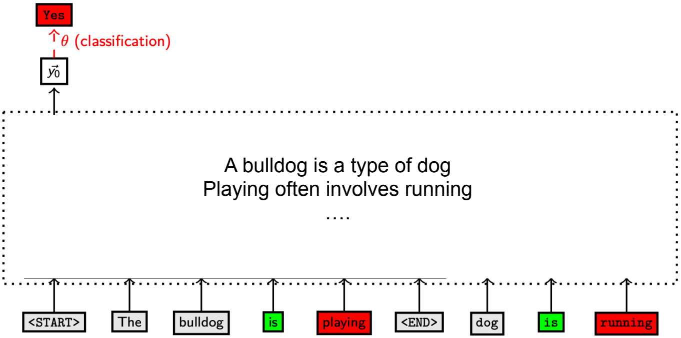
- Building specialized models on top of basic model; **classification**.



Can models *properly* learn language by reading the internet? Are they safe and reliable? *We don't have particularly good answers yet.*

Other Uses of Language Models

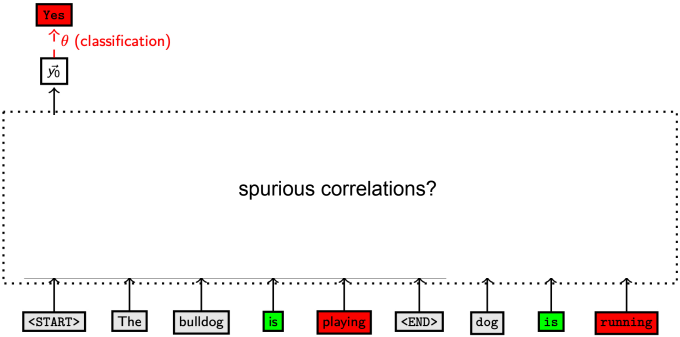
- Building specialized models on top of basic model; **classification**.



Can models *properly* learn language by reading the internet? Are they safe and reliable? *We don't have particularly good answers yet.*

Other Uses of Language Models

- Building specialized models on top of basic model; **classification**.



Can models *properly* learn language by reading the internet? Are they safe and reliable? *We don't have particularly good answers yet.*

Conclusions

- Computational modeling of language; thinking of **linguistic objects as continuous** (e.g., as vectors).
- **(Contextual) Language Models**: tools used to build representations (*"big piles of linear algebra"*); nowadays, learn by reading the internet.[3]

[3] Highly accessible publicly available tools: `https://github.com/huggingface/transformers`.

Conclusions

- Computational modeling of language ; thinking of **linguistic objects as continuous** (e.g., as vectors).
- **(Contextual) Language Models** : tools used to build representations (*"big piles of linear algebra"*); nowadays, learn by reading the internet.[3]
- Widely used throughout research and industry, high hopes about their potential; but are they safe and reliable?

[3] Highly accessible publicly available tools: `https://github.com/huggingface/transformers`.

Conclusions

- Computational modeling of language ; thinking of **linguistic objects as continuous** (e.g., as vectors).
- **(Contextual) Language Models** : tools used to build representations (*"big piles of linear algebra"*); nowadays, learn by reading the internet.[3]

- Widely used throughout research and industry, high hopes about their potential; but are they safe and reliable?

> **Why this matters:** AI technology can greatly benefit humanity, but only if it is safe; otherwise, can do serious harm.

[3] Highly accessible publicly available tools: `https://github.com/huggingface/transformers`.

Machine Learning

From [xkcd]

Would you trust a "big pile of linear algebra" to manage your finances, vet your facebook posts, help you manage your household,...?

Thank you.