Introduction to BERT

Software Engineering Decision Support Lab

University of Calgary

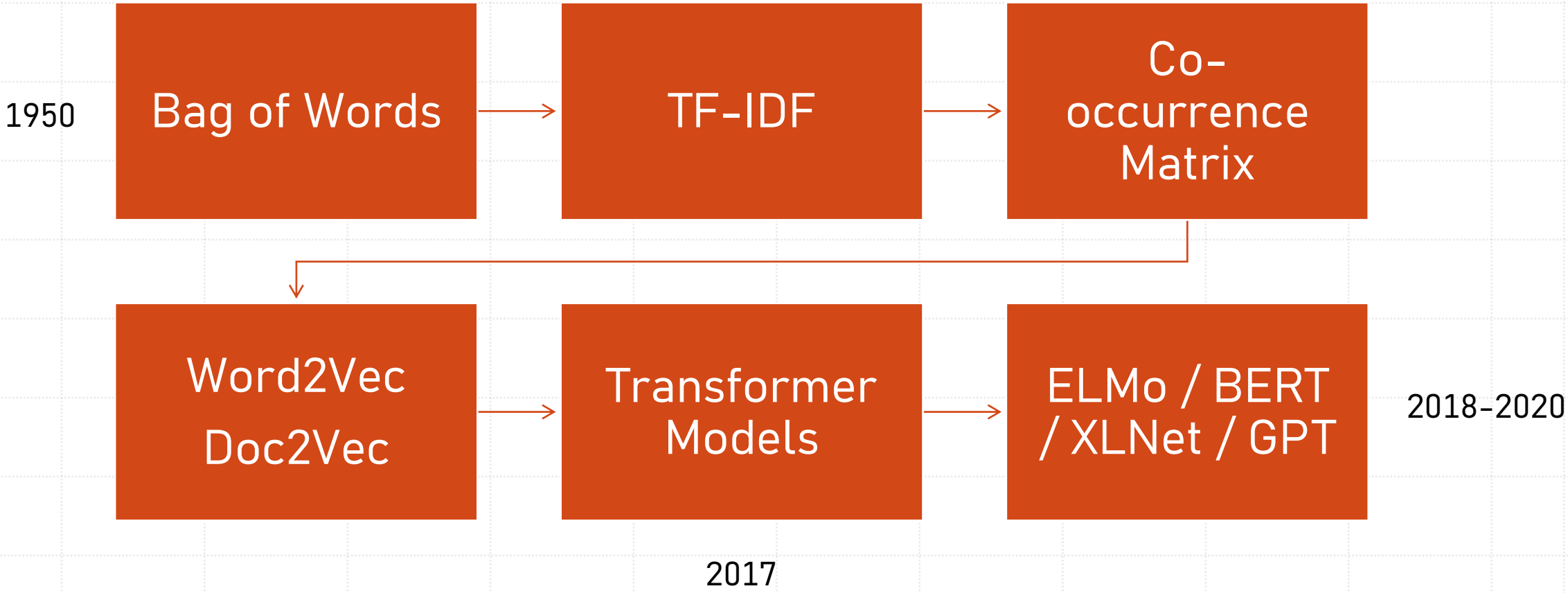
What we have covered so far

- General understanding of NLP
- Went through a full NLP workflow

Goals for today

- General understanding of BERT
- Get familiarized with steps in using BERT
- Have a fair understanding of BERT in practice

Evolution of NLP



Evolution of NLP, cont'd

- Traditionally, a language model is a **statistical model** of the probability of a sentence or phrase.
  - Bag of Words
  - TF-IDF

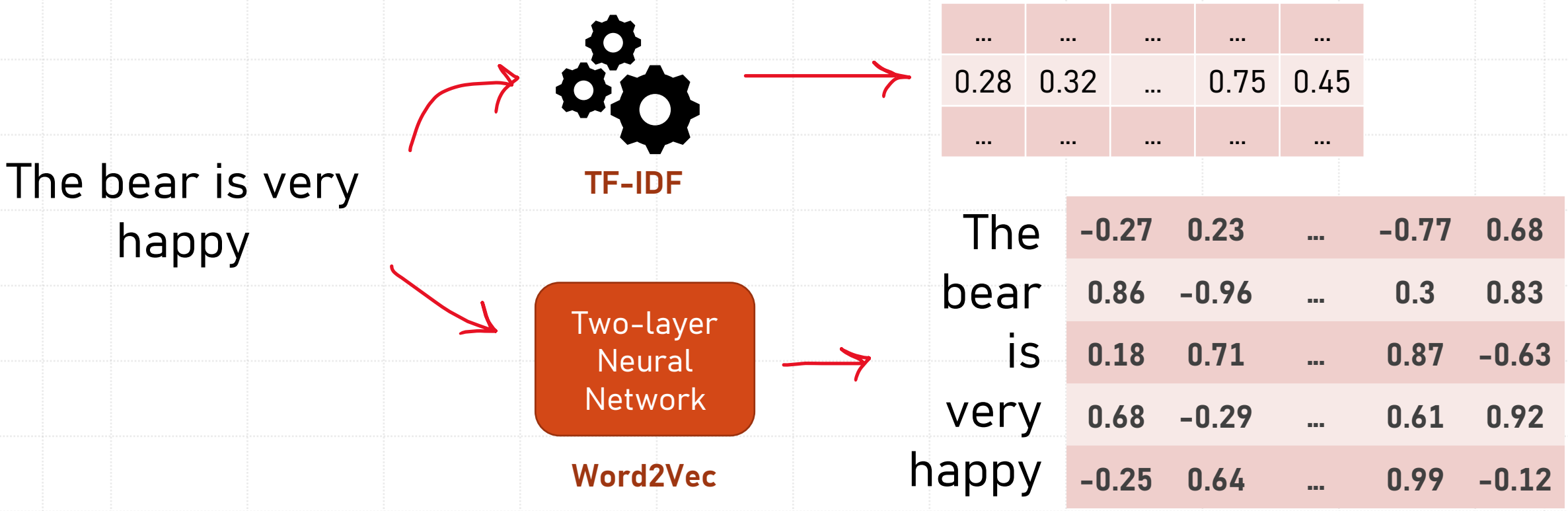
- What is the problem with these models?
  - The order of words are ignored
  - Different combination of words with different meanings are ignored (you cannot simply ignore stop words)
    - Turn right / Turn down / Turn Off / Turn out / Turn in
  - No understanding of the grammar or the language

Evolution of NLP, cont'd

- Make the models understand the language
- Build a model being able to learn language
  - Make use of Neural Networks
- Train the model on large textual corpuses, like WikiPedia or large number of Books.
- Use the pre-trained models to represent documents/sentences/words as vectors.

Word2Vec

- Why introducing Word2Vec?
- Shallow **two-layer Neural network**.
- Generates Vectors for each word; also known as **Word Embeddings**
- You shall know a word by the company it keeps.



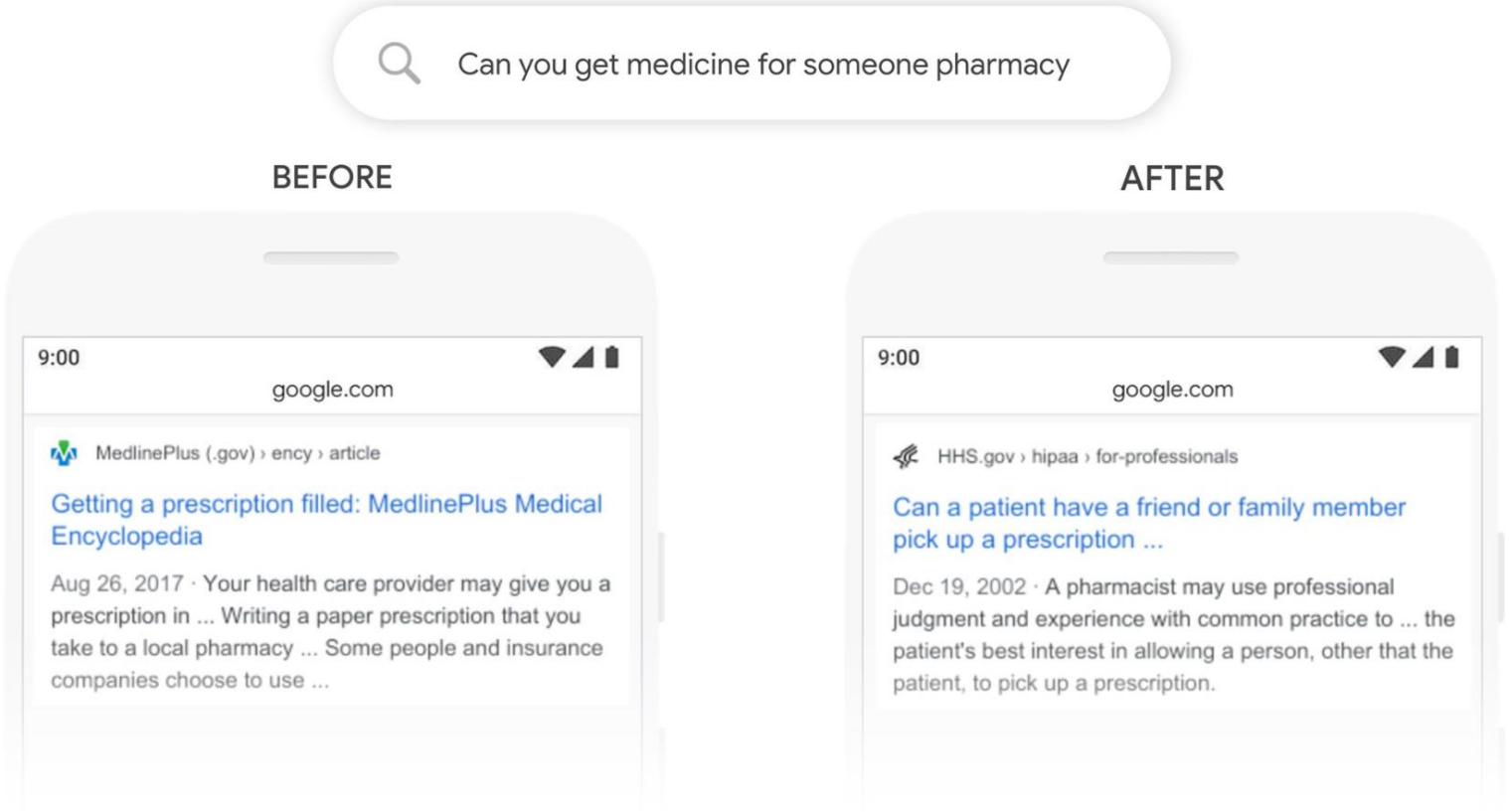
BERT

- **B**idirectional **E**ncoder **R**epresentation from **T**ransformers
- A specific, large transformer masked language model

- Before explaining any of this, lets look at **BERT** as a **pretrained black box** first and see what it does**!**

Google Search

- BERT is now used in almost all English google searches.



BERT as a black box – use cases

1. Generate word embeddings

[CLS]	I	Like	NLP	[SEP]	I	am	Learn	##ing	it	[SEP]
0.164	0.933	0.982	0.392	0.638	0.401	0.226	0.126	0.087	0.243	0.126
0.155	0.877	0.01	0.531	0.935	0.113	0.969	0.499	0.995	0.514	0.499
0.138	0.902	0.543	0.612	0.352	0.438	0.474	0.402	0.943	0.574	0.402
…	…	…	…	…	…	…	…	…	…	…
0.601	0.255	0.232	0.698	0.437	0.146	0.367	0.852	0.548	0.156	0.852
0.678	0.33	0.749	0.425	0.995	0.617	0.511	0.398	0.104	0.635	0.398
0.551	0.457	0.459	0.461	0.996	0.741	0.28	0.495	0.641	0.847	0.495

Depth 768 – 1024

New tokens:

1. CLS
2. SEP
3. ##
   - em
   - ##bed
   - ##ing
   - ##s

Further read: https://medium.com/@dhartidhami/understanding-bert-word-embeddings-7dc4d2ea54ca

1. Generate word embeddings, Cont'd

- Contains the semantics and understands meanings, not just statistics
  - Same words may have different embeddings considering where they are being used.

    Turn up / Turn right / Turn down / Turn out

    Nice day / Rainy day / Next day / 2 days

    And much more complex and deeper understanding of words and sentences.
- Heavy to process and difficult to maintain in memory

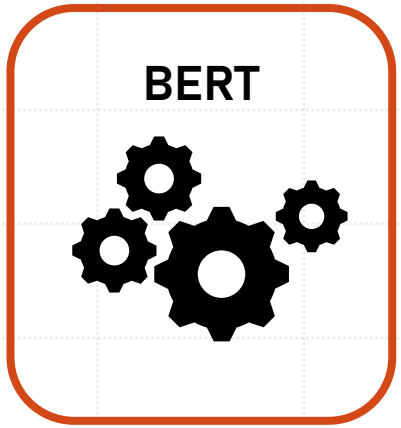
BERT as a black box – use cases, cont'd

2. Fine-tuning and classification

Labeled Data

Sentence	label
Today is a beautiful sunny day!	Joy
I hate my new phone!	Disgust
...	...
I am having a rough day!	Sadness

Fine-Tune



Prediction

Sentence	label
This tutorial is Awesome!	Joy
...	...

Only **Tokenization** and **lower casing** as pre-processing.
No **Vectorization** or other forms of **pre-processing** needed!

What BERT actually is?

- A specific, large transformer masked **language model**
- Traditionally a language models is a statistical model of the **probability of a sentence** or a phrase
- P(This tutorial is awesome) > P(tutorial this awesome is)

Source: https://www.youtube.com/watch?v=zMxvS7hD-Ug&t=1s

What BERT actually is? Cont'd

- A specific, large transformer **masked language model**
- A masked language model is trained by **removing words** and having the model **fill in the blanks**!
- Natural Language _____ is awesome!
- Masked language models are one kind of **Contextual** word embedding
  - Contextual: different representation for different senses!
  - The sentence "The dog barks" **makes sense**. The "the tree barks" **does not**!

Source: https://www.youtube.com/watch?v=zMxvS7hD-Ug&t=1s

What BERT actually is? Cont'd

- A specific, large **transformer** masked language model
- Transformers are a fairly new (2017) [1] family of Neural Network architectures
- For now, lets just think of transformers as Blackboxes that if trained well, learn from massive textual content. But here is a good short video to learn more if interested: https://www.youtube.com/watch?v=KN3ZL65Dze0

What BERT actually is? Cont'd

- A specific, **large** transformer masked language model
- The large version has 340 million trainable parameters
- It has many variations
  - BERT-Large (2018)
  - DistilBERT (2019)
  - RoBERTa
  - mBERT (Multilingual)
  - CamemBERT (French)
  - More in https://huggingface.co/
- They have to be pre-trained. VERY expensive. Just stick with the existing pre-trained versions.

Pros/cons of BERT

- Benefits:
  - Transferable Model
  - Can be have a very good accuracy
  - A lot of pre-trained models available for different use-cases in 100+ languages
- Drawbacks:
  - Big (memory), and slow to train
  - Expensive
  - Harder to implement, needs fine-tuning, can be finicky, sometimes does not converge


Enough talking,
Let's see BERT in practice

References

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30.