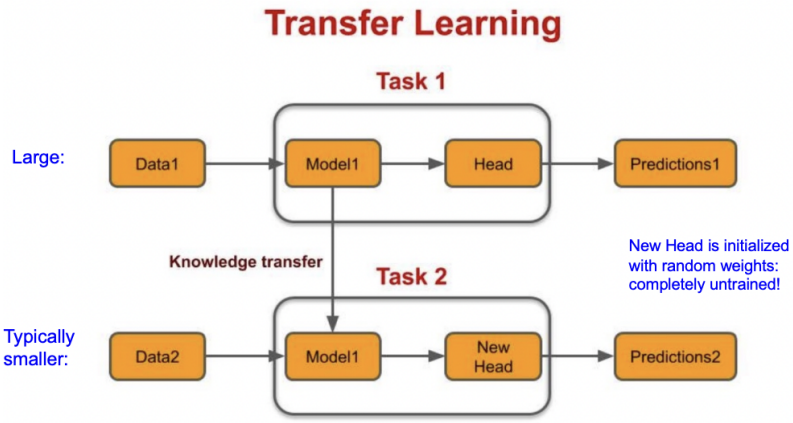
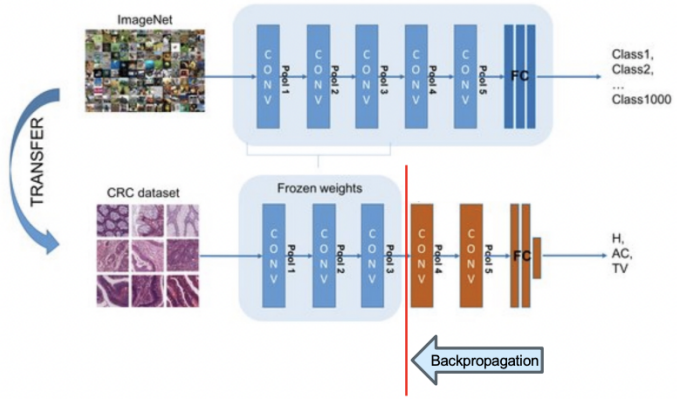
CAS CS 505
Lec 19

Presentation of LX 360/660 with Najoung Kim; The Transformer Family; BERT

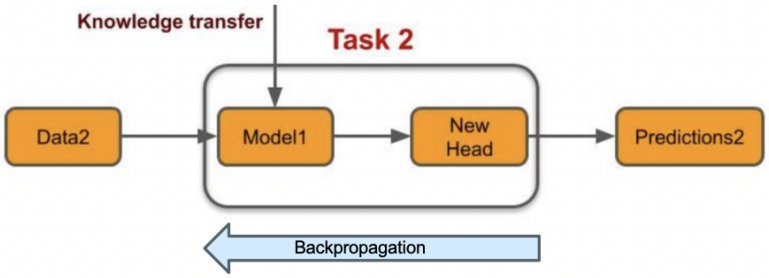
1. Paradigms for Transfer Learning



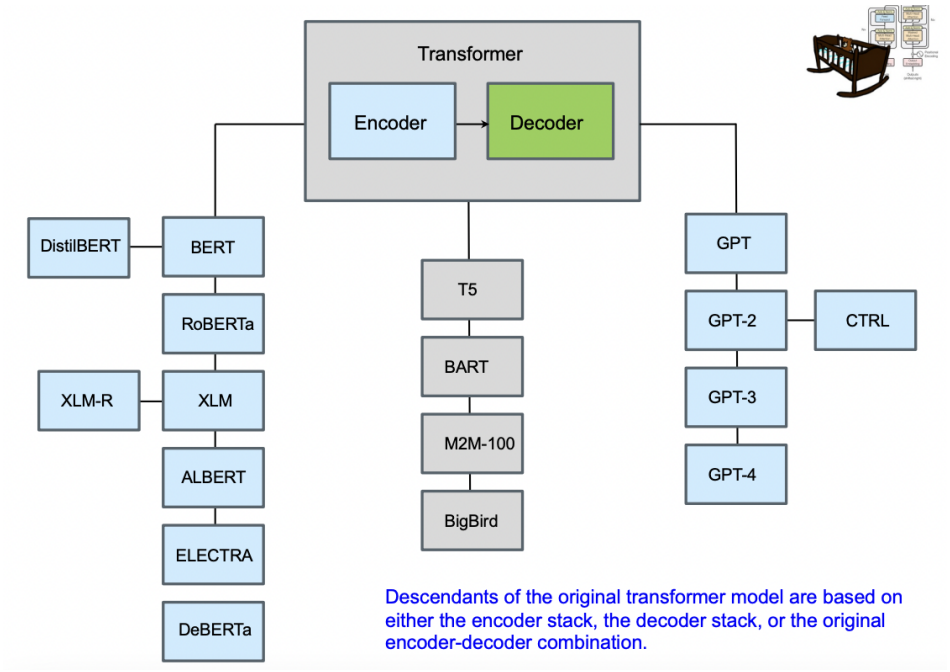
a.
b. Big question: How does backpropagation work? Do you freeze the pre-trained model or allow it to be retrained?
c. If frozen: Then you are treating the pretrained model as a feature extractor for the downstream layers:



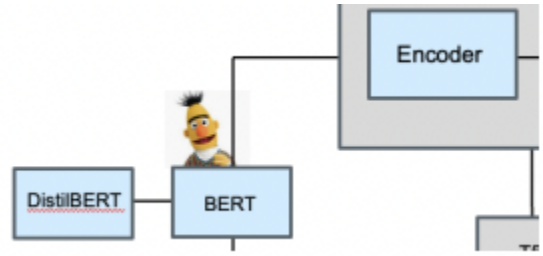
d. Big question: How does backpropropagation work?
e. If you let the model be retrained with no restrictions, e.g.,



f. Then the new head will be trained, but the new model may take a long time to train, or – much worse -- suffer from catastrophic forgetting, where the original training is disrupted and performance gets worse!
g. How to prevent catastrophic forgetting?
    i. A popular solution is Layer-wise Relevance Adjustment (LoRa)
    ii. LoRa uses Layer-wise Relevance Estimation to adjust the learning rates of the pretrained model:
        1. During pretraining:
            a. Estimate the relevance of each layer to the original task by measuring the gradients: layers with larger gradients were more relevant to the task;
        2. During retraining:
            a. Continue with LRE with respect to the new task;
            b. Use the LRE estimates to adjust how much updating should be done in these layers, or how much regularization should be performed.

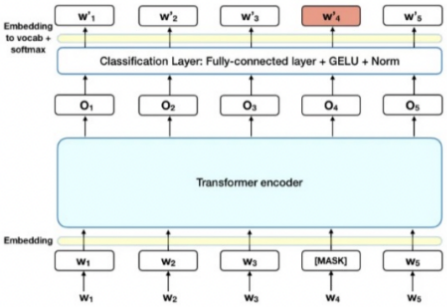
2. The Transformer Family
    a. 

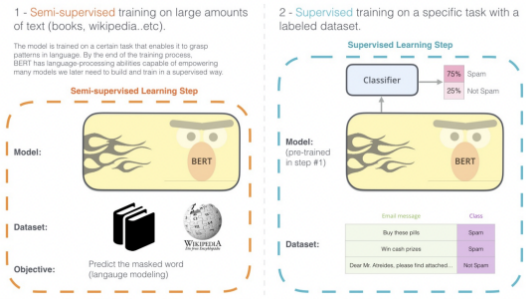
3. BERT: Bidirectional Encoder Representations from Transformers
    a. The most significant difference in the models is how they process the input sequence:



    b. BERT consists of:
        i. The stacked-encoder part of the full transformer model, with
        ii. A single linear layer on top, acting as a classifier (depending on the task);
        iii. Pretraining based on Cloze tasks and next-sentence prediction
        iv. Transfer learning to adapt to a new task.

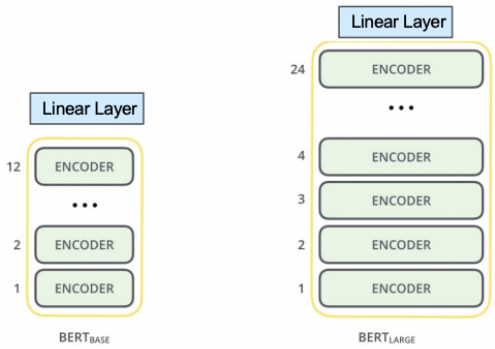
    c.





4. BERT
    a. Bert has two implementations:
        i. Bert base:
            1. 12 layers; 12 attention heads per layer;
            2. 768 hidden units; 110 M parameters
        ii. Bert large:
            1. 24 layers; 16 attention heads per layer;
            2. 1024 hidden units; 340 M parameters

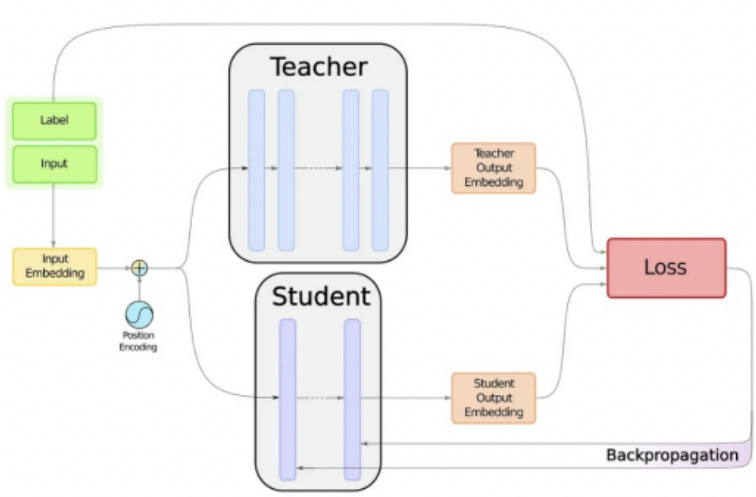
    b.



c.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

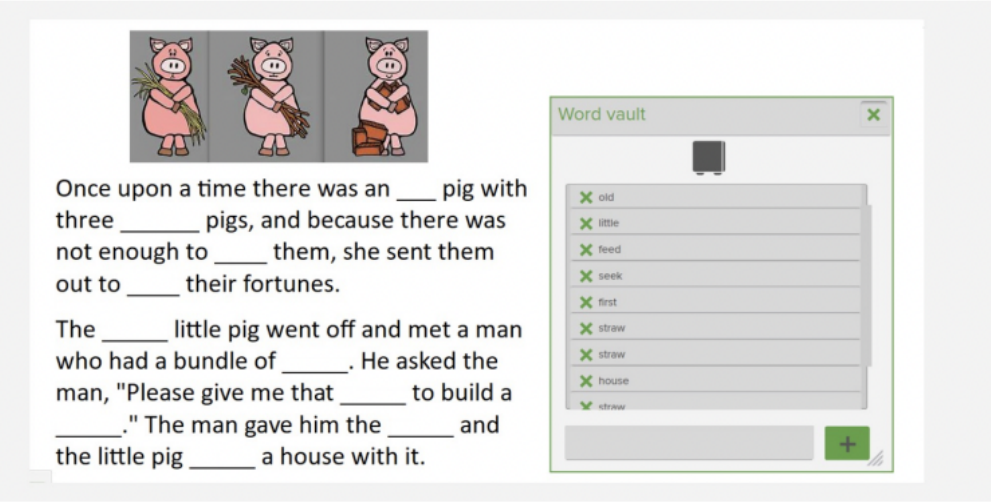
5. BERT and DistillBERT
    a. Another version, created by HuggingFace, is DistilBERT:
    b. Created using Knowledge Distillation, or Student Teacher Transfer Learning: A small network is trained not on data, but to imitate another network.
        i. BERT is the “teacher” network;
        ii. DistilBERT is a smaller “student” network:
            1. Only 6 encoder layers
            2. 40% fewer parameters
            3. 60% faster
            4. Achieves 97% of BERT’s performance.



iii.

6. BERT: Training
    a. BERT added two significant ideas for training which allowed it to achieve SOTA performance on significant tasks:
    b. Training using a Masked Language Model (MLM)
    c. Focus on a "next sentence prediction" task with two sentences as input.
    d. Note: The first versions of BERT used the 800 million word BooksCorpus and a 2.5 B word English Wikipedia corpus.
    e. The Masked Language Model is based on an educational theory/testing paradigm known as the Cloze Task, where students learn a language by filling in blanks in a

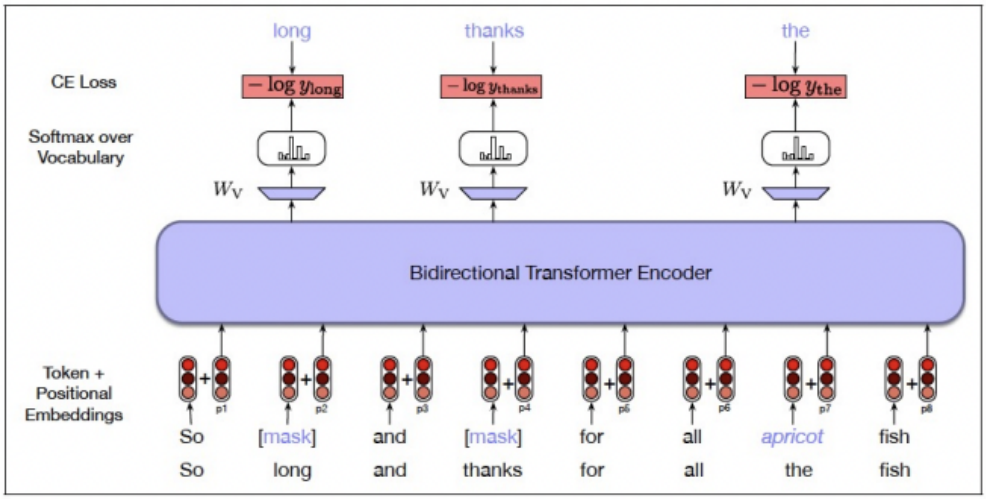
story or piece of text:



f. Masked Language Modeling uses unannotated text from a large corpus. 15% of the words in the corpus are selected for the training phase: of these,
   i. 80% of replaced with the token [MASK]
   ii. 10% are replaced with randomly-selected tokens
   iii. 10% are left unchanged.

g. The model is trained to predict the missing tokens.

h.



i. A variation of MLM uses spans (subsequences of the input sequence); all of the words in the span are replaced as before:

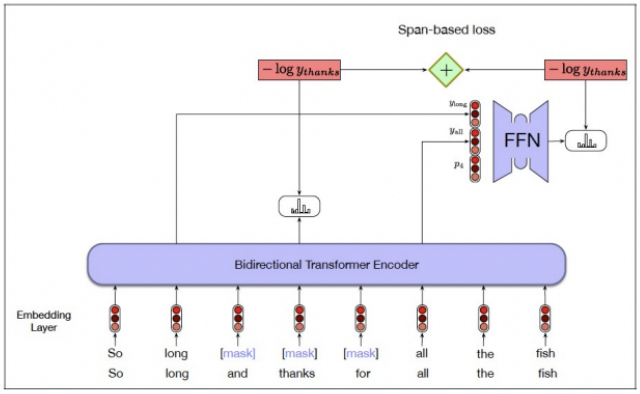


Figure 11.6 Span-based language model training. In this example, a span of length 3 is selected for training and all of the words in the span are masked. The figure illustrates the loss computed for word *thanks*; the loss for the entire span is based on the loss for all three of the words in the span.

j. The Next Sentence Prediction task is to input TWO sentences starting with the token [CLS] and separated by the token [SEP]. The training set has 50% sentences that are next to each other in the corpus, and 50% random sentences.

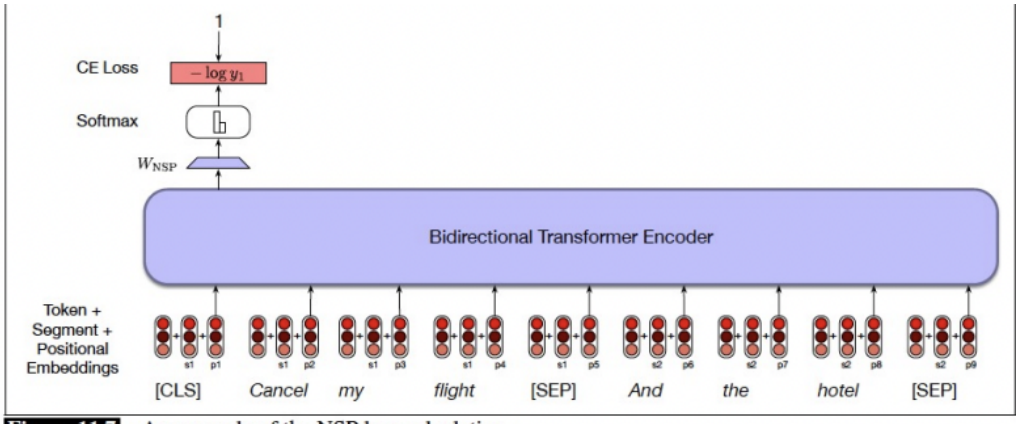


Figure 11.7 An example of the NSP loss calculation.

k. Bert can be used for sentence classification if a single sentence is input:

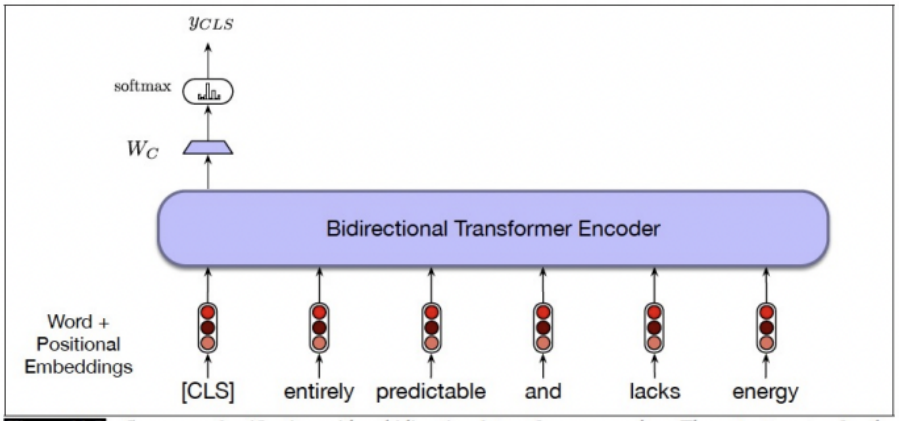


Figure 11.8 Sequence classification with a bidirectional transformer encoder. The output vector for the [CLS] token serves as input to a simple classifier.

l. BERT can classify the relationship between two sentences:

- `Neutral`
  - a: Jon walked back to the town to the smithy.
  - b: Jon traveled back to his hometown.
- `Contradicts`
  - a: Tourist Information offices can be very helpful.
  - b: Tourist Information offices are never of any help.
- `Entails`
  - a: I'm confused.
  - b: Not all of it is very clear to me.

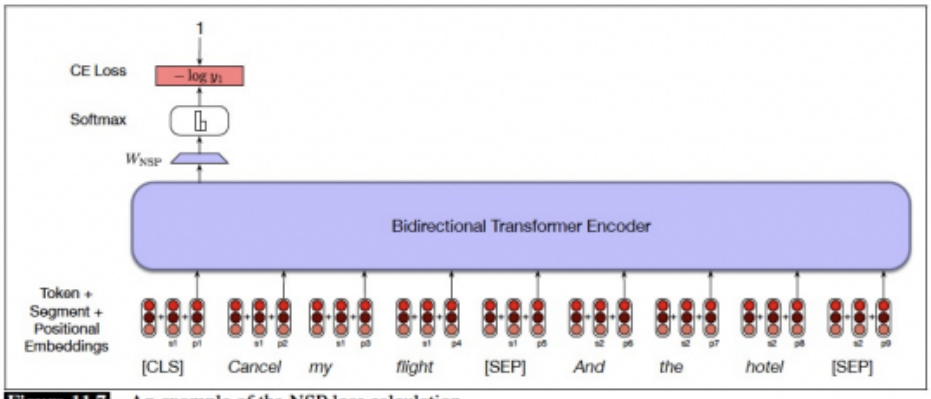


Figure 11.7 An example of the NSP loss calculation.

m. BERT can generate the most likely sentence to follow a given sentence:

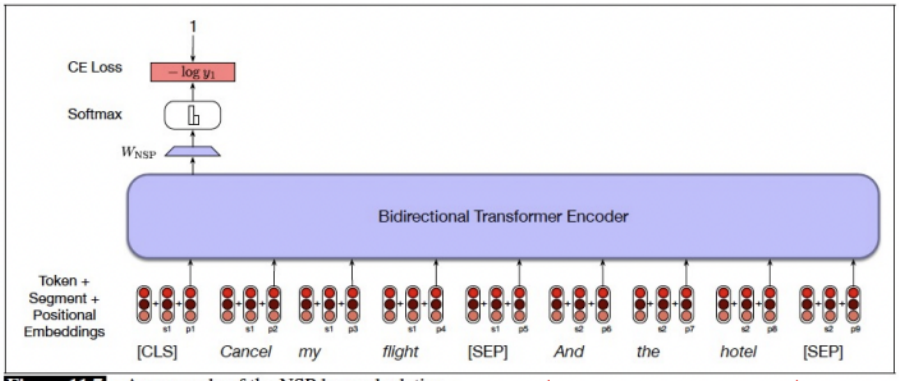


Figure 11.7 An example of the NSP loss calculation.

Use Beam Search to find most likely sentence to follow.

7. Using BERT
    a. Bert can be used for sequence labelling if all of the outputs are used:

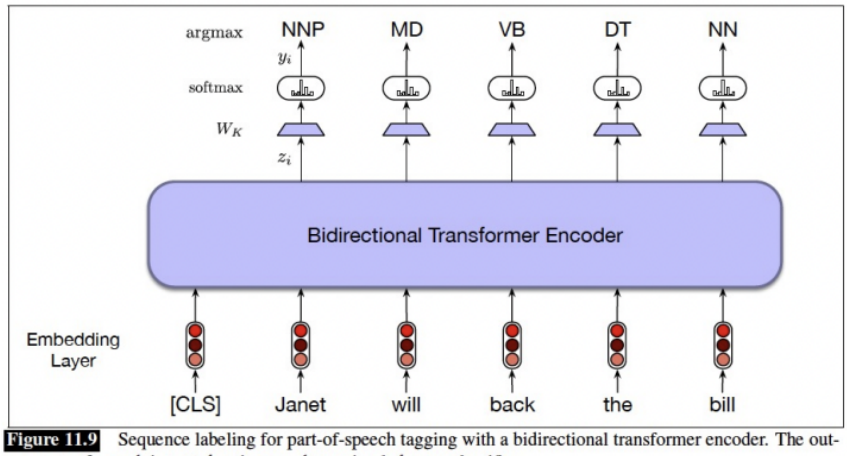
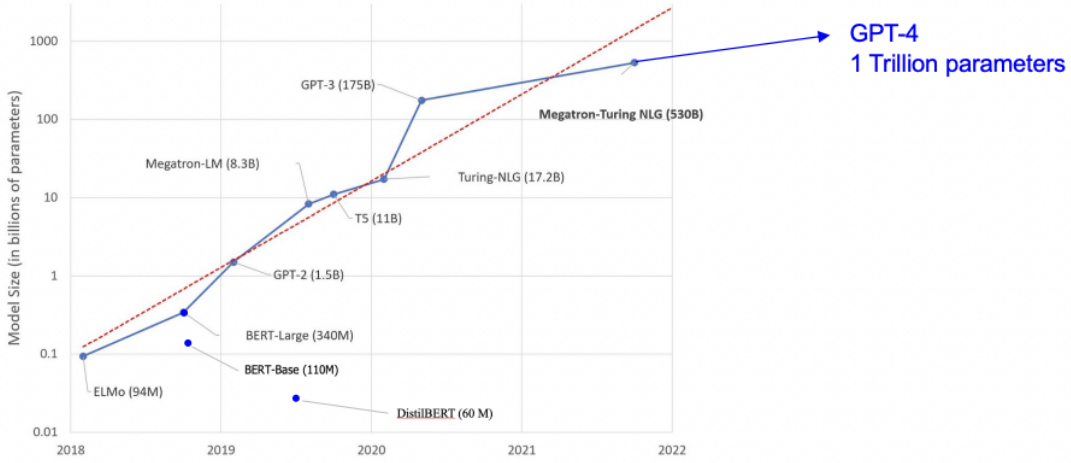


Figure 11.9 Sequence labeling for part-of-speech tagging with a bidirectional transformer encoder. The output vector for each input token is passed to a simple k-way classifier.

8. BERT Punches Above Its Weight

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**



    a.

9. Most Important NLP Tasks: Classification
    a. Sentiment Analysis: identifying the position of a piece of text in some scale of sentiment.
    b. Position may be categorical (2 stars out of 5) or continuous in some range (2.3 on a scale 0 .. 10)
    c. Types of sentiment:
        i. Positive – Negative
        ii. Aspect or point of view or bias (e.g., political)

iii. Intent detection
iv. Emotion Detection
    1. Happiness
    2. Excited/enthusiastic
    3. Frustration or Anger
v. Friendship, affection, love or sexual attraction
vi. Humorous
vii. Irony
viii. Hate speech and Fake News detection (next slide)

10. Fake News and Hate Speech Detection
    a. Fake News Detection: detecting and filtering out texts containing false and misleading information.
    b. Stance Detection: determining an individual's reaction to a primary actor's claim. It is a core part of a set of approaches to fake news assessment.
    c. Hate Speech Detection: detecting if a piece of text contains hate speech.

Why is NLP so necessary in this?

- Users watch 4,146,600 YouTube videos
- 456,000 tweets are sent on Twitter
- Instagram users post 46,740 photos

With 2 billion active users Facebook is still the largest social media platform. Let that sink in a moment—more than a quarter of the world's 7 billion humans are active on Facebook! Here are some more intriguing Facebook statistics:

- 32 billion people are active on Facebook **daily**
- Europe has more than 307 million people on Facebook
- There are five new Facebook profiles created every second!
- More than 300 million photos get uploaded per day
- Every minute there are 510,000 comments posted and 293,000 statuses updated

1.836 Billion Facebook Posts each day....

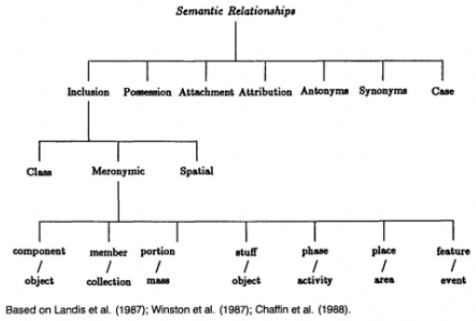
Even though Facebook is the largest social network, Instagram (also owned by Facebook) has shown impressive growth. Here's how this photo-sharing platform is adding to our data deluge:

- There are 600 million Instagrammers; 400 million who are active every day
- Each day 95 million photos and videos are shared on Instagram
- 100 million people use the Instagram "stories" feature daily

    d.

11. Information Retrieval
    a. (An old subject, even before Google made it the the most popular text-processing task.)
    b. Resource Retrieval from text queries/questions
        i. Resource could be
            1. Highly structured (relational database, code)
            2. Semi-structured (Markup Languages (XML), labeled documents)
            3. Unstructured (documents)

    ii. Database search from keywords
    iii. Google search
    iv. Backend to Speech to Text systems (siri)
    v. Question Answering (next slide)
  c. Sentence/document similarity: determining how "similar" two texts are
    i. Notion of "similar" is variable (similar topic, similar sentiment, ...)
    ii. Relationship to IR:
      1. How similar is text query to a document?
      2. "Retrieve more documents similar to this one"
    iii. Create a map/graph of documents similar to given sentence/document
    iv. Plagiarism/copyright infringement
  d. Document Ranking: Rank documents as to some criterion (e.g., PageRank)
    i. How well does this document satisfy my query?
    ii. How important/authoritative is this document?
12. Entities, Relations, and Knowledge Graphs
  a. Named Entity Recognition: tagging entities in text with their corresponding type, typically in BIO notation.)
  b. Coreference Resolution: clustering mentions in text that refer to the same underlying real-world entities.
  c. Relation extraction: extracting semantic relationships from a text, e.g.,
    i. Is-A
    ii. Has-A
    iii. Son-Of
    iv. Part-Of
    v. Size-of
    vi. etc., etc., etc.
  d. Build a graph structure:
    i. Knowledge Graph
    ii. Concept Map
    iii. Mind Map
  e. Graphs can be used to enhance other NLP tasks: search, similarity, question answering, etc
  f. Entity Linking: recognizing and disambiguating named entities to a knowledge base (e.g., Wikidata).
  g. Relation prediction: identifying a named relation between two named o semantic entities.



Based on Landis et al. (1987); Winston et al. (1987); Chaffin et al. (1988).

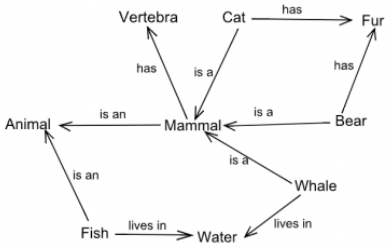


Fig. 1. Sample knowledge graph. Nodes represent entities, edge labels represent types of relations, edges represent existing relationships.

h.

13. Text-To-Text Generation
    a. Machine Translation: translating from one language to another.
        i. Covered in lecture – Transformer technology transformed this task
    b. Text Generation: creating text from a prompt or subject phrase that appears indistinguishable from human-written text.
        i. Covered in lecture – Use language models, Large Language Models (GPT) have transformed this task
    c. Lexical Normalization: translating/transforming a non-standard text to a standard register.
    d. Paraphrase Generation: creating an output sentence that preserves the meaning of input but includes variations in word choice and grammar.
    e. Text Simplification: making a text easier to read and understand, while preserving its main ideas and approximate meaning.

How Large Language Models are Transforming Machine-Paraphrased Plagiarism

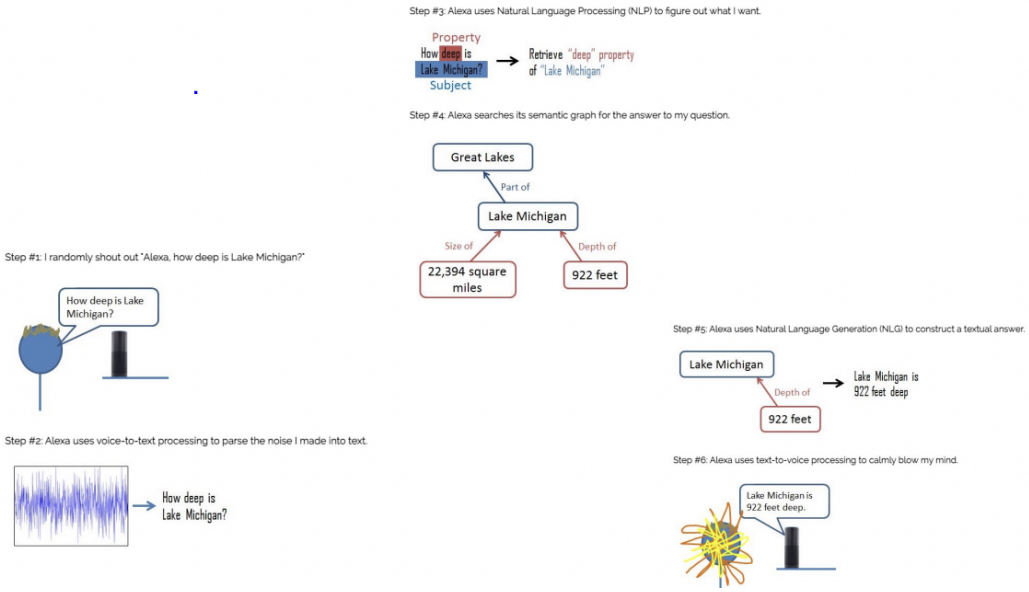
Jan Philip Wahle[♣*], **Terry Ruas**[*], **Frederic Kirstein**[♠*], **Bela Gipp**[*]
[*]Georg-August-Universität Göttingen, Germany
[♠]Mercedes-Benz Group AG, Germany
[♣]wahle@gipplab.org

f.

g. Text Summarization (next slide)

14. Topics and Keywords; Text Summarization
    a. Topic Modeling: identifying abstract “topics” underlying a collection of documents.
    b. Keyword Extraction: identifying the most relevant terms to describe the subject of a document
    c. Text Summarization: Reducing size of document while preserving the most important information
        i. Extractive:
            1. Identify the most important sentences in a document and construct the summary from these exact sentences
            2. TextRank, LexRank (implements PageRank on sentences in a document)
            3. Latent Semantic Analysis (Singular Value Decomposition on a wordsentence matrix)
        ii. Abstractive:
            1. Create new text summarizing main points
            2. Use of Large Language Models: GPT, BERT, etc...
        iii. Use cases for Text Summarization:
            1. Summaries for busy executives or (students!)
            2. Summaries of articles, books, chapters
            3. Automatic Table of Contents or Indices
            4. Downstream from Speech-to-Text systems:
                a. Notetaking of meetings, lectures
                b. Abstracts of podcasts, YouTube videos
                c. Automatic summary of customer phone calls
15. Chatbots and Question Answering
    a. Slot Filling or Cloze Task: aims to extract the values of certain types of attributes (or slots, such as cities or dates) for a given entity from texts.
    b. Chatbots: Conversation agents (started with Eliza in early 1060's!)
    c. Dialog Management: managing of state and flow of conversations.
    d. Question Answering: Responding to textual queries with textual answers
        i. Extractive QA: The model extracts the answer from a knowledge source, such as a knowledge graph, database, or document (next slide).
        ii. Open Generative QA: The model generates free text directly based on the (global) context.
        iii. Closed Generative QA: The model generates free text directly based only on the question.

16. Question Answering Using Knowledge Graphs



a.

17. Reasoning with Text
    a. Logical Relationship of two sentences/documents:
        i. Entailment
        ii. Temporal sequence
        iii. Specialization
    b. Subsystem of text generation at scale
    c. Text-to/from-First Order Logic: Translate between text and expressions in first-order logic:

No student failed Chemistry, but at least one student failed History.
$\neg\exists x\,(\text{Student}(x) \wedge \text{Failed}(x,\text{Chemistry})) \wedge \exists x\,(\text{Student}(x) \wedge \text{Failed}(x,\text{History}))$

    d. Use cases:
        i. Teaching logic
        ii. Game/puzzle solving
        iii. Interface to automated theorem prover
            1. Prolog
            2. Planner
            3. Wolfram Alpha

18. Text-to-Data and Data-to-Text
    a. Text-to-Image: generating photo-realistic images which are semantically consistent with the text descriptions.
    b. Image captioning: Generate captions for input images
    c. Video-to-Text: Generating text describing a sequence of images
    d. Text-to-Speech: Human-like reading of input text.

e. Speech-to-Text: transcribing speech to text

An example of some of the images created by Imagen, Google's text-to-image AI generator.