Lesson 4: Word Embeddings and Transformer Models

🔍 1. The Problem with Traditional Text Representations

In the early days of NLP, computers processed text in a very naïve way. Words were simply treated as discrete symbols—nothing more than individual IDs in a vocabulary.

This was problematic because:

- It ignored **semantic relationships** (e.g., “king” and “queen” were as unrelated as “king” and “toaster”).
- Models couldn’t **generalize** well to new or unseen words.
- High-dimensional sparse representations like **one-hot vectors** were memory inefficient and lacked meaning.

🌐 2. Word Embeddings: Giving Meaning to Words

Word embeddings represent words as dense vectors in a continuous vector space, where similar words lie close to one another. These vectors capture syntactic and semantic meaning based on context.

Popular Word Embedding Models:

1. Word2Vec (Google)
    - Learns word associations from a large corpus.
    - Two main architectures:
        - Skip-gram: Predicts surrounding words given a target word.
        - CBOW (Continuous Bag of Words): Predicts a target word from surrounding words.
2. GloVe (Stanford)
    - Stands for “Global Vectors for Word Representation.”
    - Combines the benefits of global matrix factorization and local context-based learning.
3. FastText (Facebook)
    - Enhances Word2Vec by using subword information (n-grams), making it better at handling rare and out-of-vocabulary words.

💡 3. Why Word Embeddings Matter

- Semantics in Geometry: In embedding space, we can perform analogies like: king - man + woman ≈ queen
- Similarity: Words like “cat” and “kitten” have closer vectors than “cat” and “car.”
- Efficiency: Dense vectors are much more compact and informative than sparse representations.

Word embeddings changed how machines "understand" language by giving numerical meaning to words, but they had one big limitation: each word had only one vector, regardless of context.

⚡ 4. The Contextual Breakthrough: Transformers

The next revolution came with transformers, which allowed for contextual word representations. Now, the meaning of a word could change depending on its context.

Example

- In “He opened the bank account,” and “She sat on the river bank,” the word “bank” has two completely different meanings. Transformers can capture this.

---

5. Understanding the Transformer Architecture

Introduced in 2017 in the seminal paper "Attention Is All You Need," the transformer model became the foundation of modern NLP.

Key Features:

- Self-Attention Mechanism:
  Allows the model to look at all words in a sentence at once and weigh their importance when understanding a specific word. This is what gives transformers their power to model long-range dependencies.
- Parallelization:
  Unlike RNNs (which process sequentially), transformers process words in parallel, greatly speeding up training.
- Scalability:
  Easy to scale up with more layers and data.

Core Components:

- Encoder-Decoder structure (original transformer architecture)
- Positional Encoding (adds order to word sequences)
- Multi-Head Attention (captures different aspects of meaning)
- Feedforward layers and residual connections

6. BERT: Bidirectional Encoder Representations from Transformers

Developed by Google in 2018, BERT marked a shift in NLP modeling strategies:

- Bidirectional Understanding:
  Instead of looking left-to-right or right-to-left, BERT looks in both directions simultaneously. This allows it to understand full sentence context.
- Pre-training & Fine-tuning Paradigm:
  1. **Pre-training:** BERT is trained on large corpora using self-supervised tasks like:
     - Masked Language Modeling (predicting masked words)
     - Next Sentence Prediction
  2. **Fine-tuning:** The pre-trained BERT can then be adapted to specific tasks (e.g., sentiment analysis, question answering).

BERT's Impact:

- State-of-the-art results in 11 NLP tasks upon release.
- Hugely popular in both research and industry applications.

---

7. Other Transformer-Based Models

After BERT's success, many variants and successors were introduced:

- **GPT (Generative Pre-trained Transformer)** – Developed by OpenAI
  - Autoregressive model (good for text generation).
  - Powers ChatGPT.
- **RoBERTa (Facebook)** – Robustly optimized BERT with more data and training.
- **DistilBERT** – Smaller, faster BERT with minimal performance loss.
- **T5 (Text-To-Text Transfer Transformer)** – Treats all tasks as text generation problems.
- **XLNet, ALBERT, ELECTRA** – Each introduces novel improvements on BERT's architecture or training.

🔁 8. Comparing Traditional Embeddings vs. Transformers

Feature	Word2Vec / GloVe	BERT / Transformers
Context Awareness	❌ Static vectors	✅ Contextualized embeddings
Task Specificity	❌ Generic only	✅ Fine-tunable for tasks
Model Complexity	✅ Lightweight	❌ Computationally heavy
Performance	⚠️ Limited in complex tasks	✅ State-of-the-art
Sentence-Level Meaning	❌ No	✅ Yes

---

9. Why This Matters in the Real World

Thanks to word embeddings and transformers:

- Virtual assistants (like Siri, Alexa) can understand nuanced questions.
- Search engines deliver more relevant results.
- Chatbots can hold more meaningful conversations.
- Translation and summarization tools are now impressively accurate.

📝 10. Key Takeaways

- Word embeddings (Word2Vec, GloVe) encode words as dense vectors, capturing meaning and similarity.
- Transformers introduced context into word representations, revolutionizing NLP.
- BERT and its descendants now power state-of-the-art language applications.
- Transformer models have become the new gold standard in modern NLP pipelines.