UNIVERSITY OF SURREY

People-Centred AI
UNIVERSITY OF SURREY

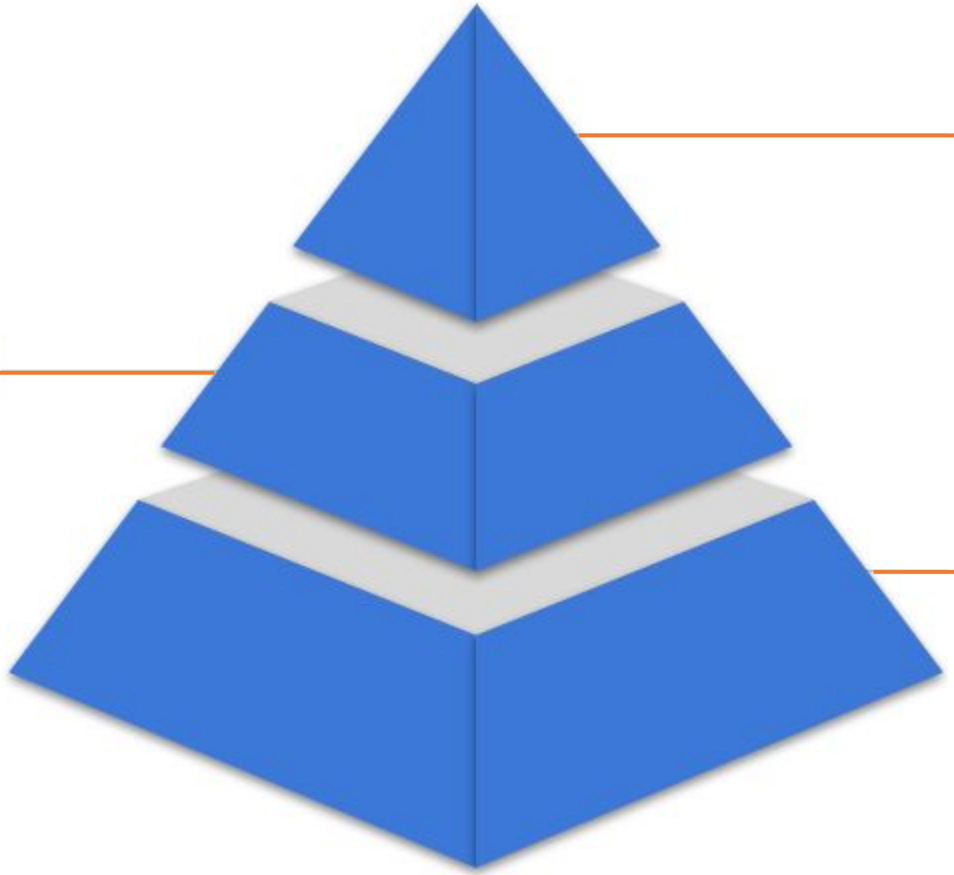
Transformers

Perspectives from Natural Language Processing (NLP)

Dr Diptesh Kanojia
(https://dipteshkanojia.github.io)

Surrey Institute for People-Centred AI
Department of Computer Science, University of Surrey

Natural Language Processing (NLP): Goal Perspective



Analyse Human Language

Textual analytics, extraction, and retrieval to analyze the information present in human language.

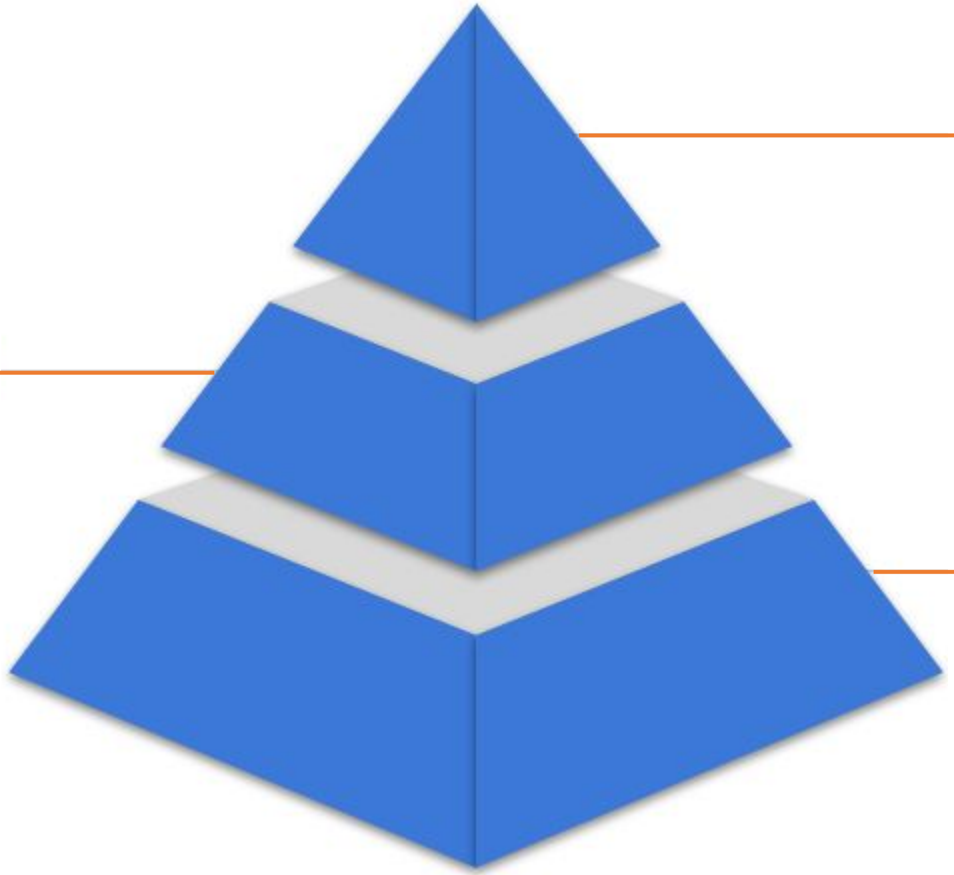
Generate Human Language

Generation of understandable human language to interface with humans.

Understand Human Language

A key goal of NLP is to ensure that machines understand human language.

Natural Language Processing (NLP): Task Perspective



3

Generate Human Language

- Machine **Translation**
- Text **Summarization** (incl. Extreme)
- Language Generation Tasks
- **Image** Captioning
- **Audio** Description & <u>many more.</u>

2

Analyse Human Language

- **Sentiment** Analysis
- **Emotion** Recognition
- **Entity** Recognition & Linking
- **Acronym/Abbreviation** Extraction

.
.
.

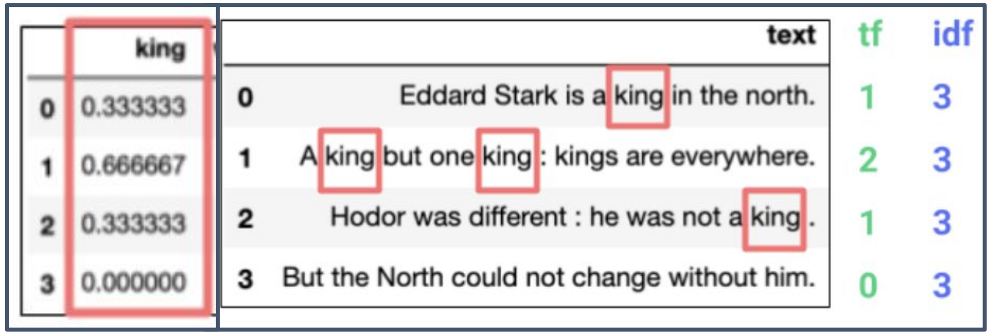
1

Understand Human Language

- **Encoding text into mathematical representations**
- **Sense** Disambiguation
- Base of other NLP tasks.
- **Cognitive NLP**

Encoding Paradigm: Evolution

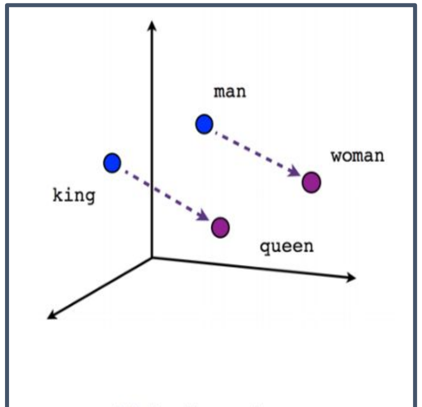
- 1 - hot encoding
- Term Frequency - Inverse Document Frequency (TF-IDF)
  - Based on ‘term’ counts*, i.e.,* frequency in the sentence and its frequency in the ‘document’



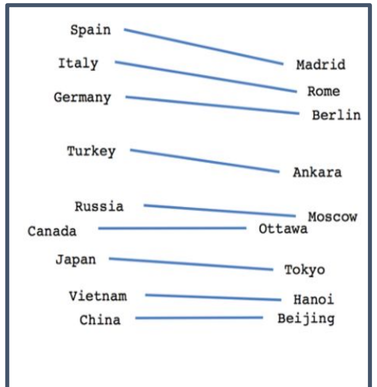
	king
0	0.333333
1	0.666667
2	0.333333
3	0.000000

	text	tf	idf
0	Eddard Stark is a king in the north.	1	3
1	A king but one king : kings are everywhere.	2	3
2	Hodor was different : he was not a king .	1	3
3	But the North could not change without him.	0	3

- Word Vectors / Embeddings
  - TF-IDF does not take into account the contextual presence of the word in a document.
  - Word embeddings use an unsupervised approach to project the word into an ‘n’-dimensional space allowing vector operations for complex tasks.
    - V(King) - V(Man) + V(Woman) = V(Queen)
    - Madrid:Spain::Rome:?
  - However, capturing ‘semantics’ requires the true context of a word across multiple senses.



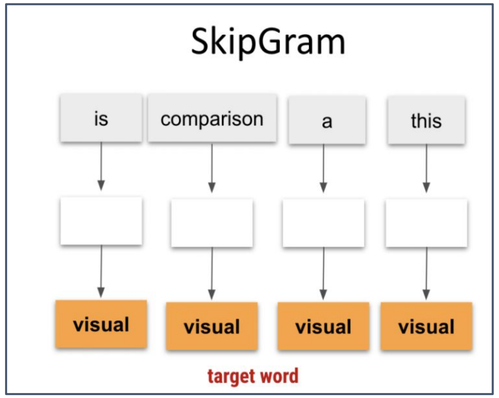
Male-Female



Country-Capital

Vectorization Approaches

- word2vec (Mikolov et. al., 2013)
  - First implementation of embeddings words or ‘tokens’ given a large monolingual corpus, i.e., a document containing a set of sentences in a single language.
  - Significant push to the NLP research sub-area.
- fastText (Bojanowski et. al., 2017)
  - Enriched word vectors with subword information.
    - Can help tackle morphology related issues.
  - Significant push to Indian language NLP, Multilingual approaches.
- MUSE (Conneau et. al., 2019) / VecMap (Artetxe et. al., 2019)
  - Approaches to build embedding models for cross-lingual / bilingual word embeddings using projection methodologies.



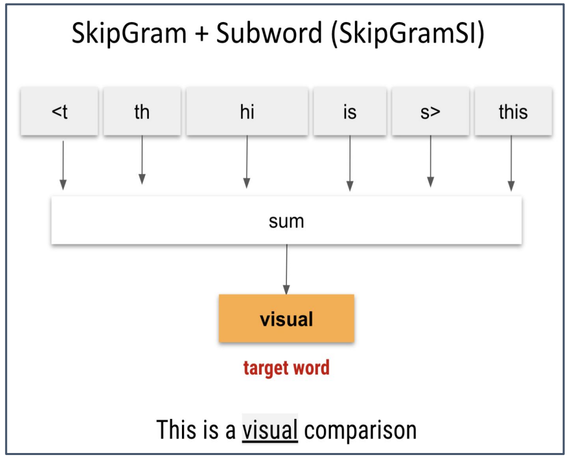


Image source: https://kavita-ganesan.com/fasttext-vs-word2vec/

Some more vectorization approaches

GloVe (Pennington et. al., 2014)

- Global Vectors for Word Representations: constructs a large matrix of (words x context) co-occurrence information, *i.e.,* for each ‘word’ (the rows), count how frequently this word is in some “context” (the columns)
- then, factorize this matrix to yield a lower-dimensional (word x features) matrix, where each row now yields a vector representation for the corresponding word/token.

Flair (Akbik et. al., 2018) [post-BERT]

- Contextualized string embeddings based on character sequences taken into account during training
- Leverages the internal states of a trained character language model.
- Distinct properties that they
  - are trained without any explicit notion of words and thus model words as sequences of characters, and
  - are contextualized by their surrounding text, meaning that the same word will have different embeddings depending on its contextual use.

The Transformer Revolution: BERTology!

BERTology

- **Encoders**: BERT, DistilBERT, RoBERTa, ALBERT, DeBERTa, ELECTRA (discriminator), Longformer, …
- **Multilingual** Encoding: XLM, XLM-R, mBERT, IndicBERT, MuRIL, …
- **Decoders** (Autoregressive): XLNet, GPT-n, Reformer, OPT
- **Decoders** (Non-autoregressive): CoMMA, DisCo, CMLMC, Levenshtein Transformer, PNAT
- **Encoder-Decoder**: BART, PEGASUS, T5, mT5 (multilingual), mBART (multilingual), IndicBART(multilingual),
- **Contrastive Learning Objective**: Sentence-BERT, Sentence-RoBERTa, …
    - Siamese Network like objective function, triplet loss
- **Domain-specific**: FinBERT, SciBERT, SportsBERT, Legal-BERT, BioBERT…
- **Language-agnostic**: LASERn

BERT (<u>B</u>idirectional <u>E</u>ncoder <u>R</u>epresentations from <u>T</u>ransformers)

(NLPs ImageNet moment!)

BERT and BERT-like architectures belong to the family of *autoencoding computational models* that provide vectors/embeddings for word(s)/sentences.

Built **on top of a lot of ideas**:

Semi-supervised Sequence Learning (Andrew Dai, Quoc Le) [<u>Learning Objective via Masking</u>]

ELMo (Peters et. al.) [<u>Contextual Embeddings</u>]

ULMFiT (Howard and Ruder) [<u>Transfer Learning</u>]

OpenAI Transformer (Radford et. al.) [w/ Sutskever] [<u>Decoder</u>]

Transformer (Vaswani et. al.) [<u>Core Model</u>]

Enables <u>transfer learning</u> - prime reason for BERT use.

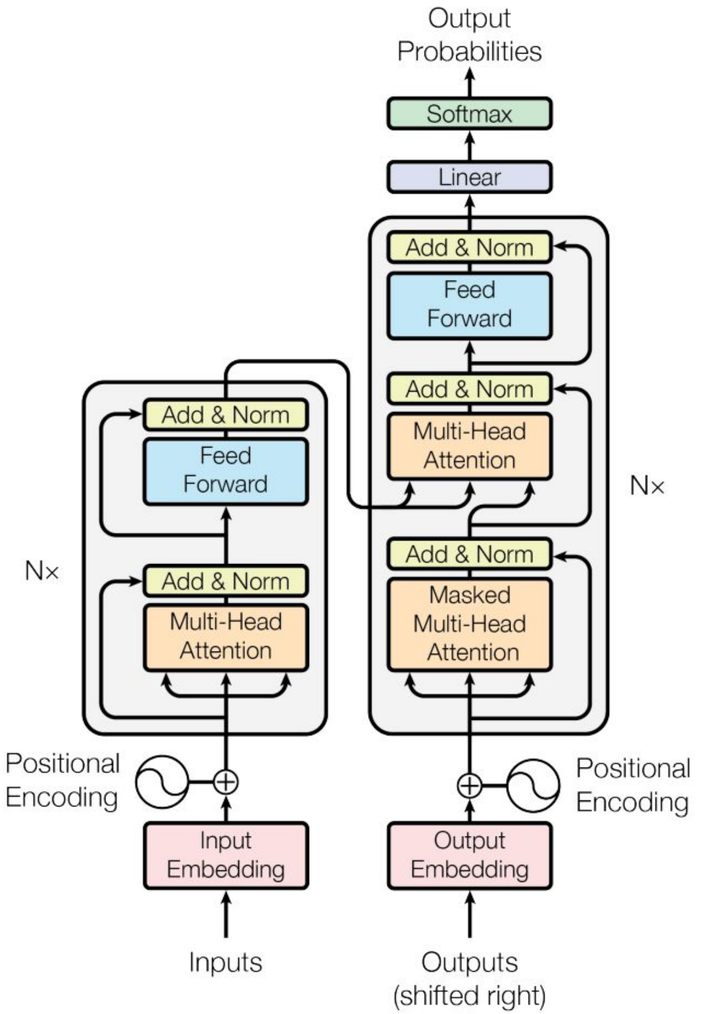
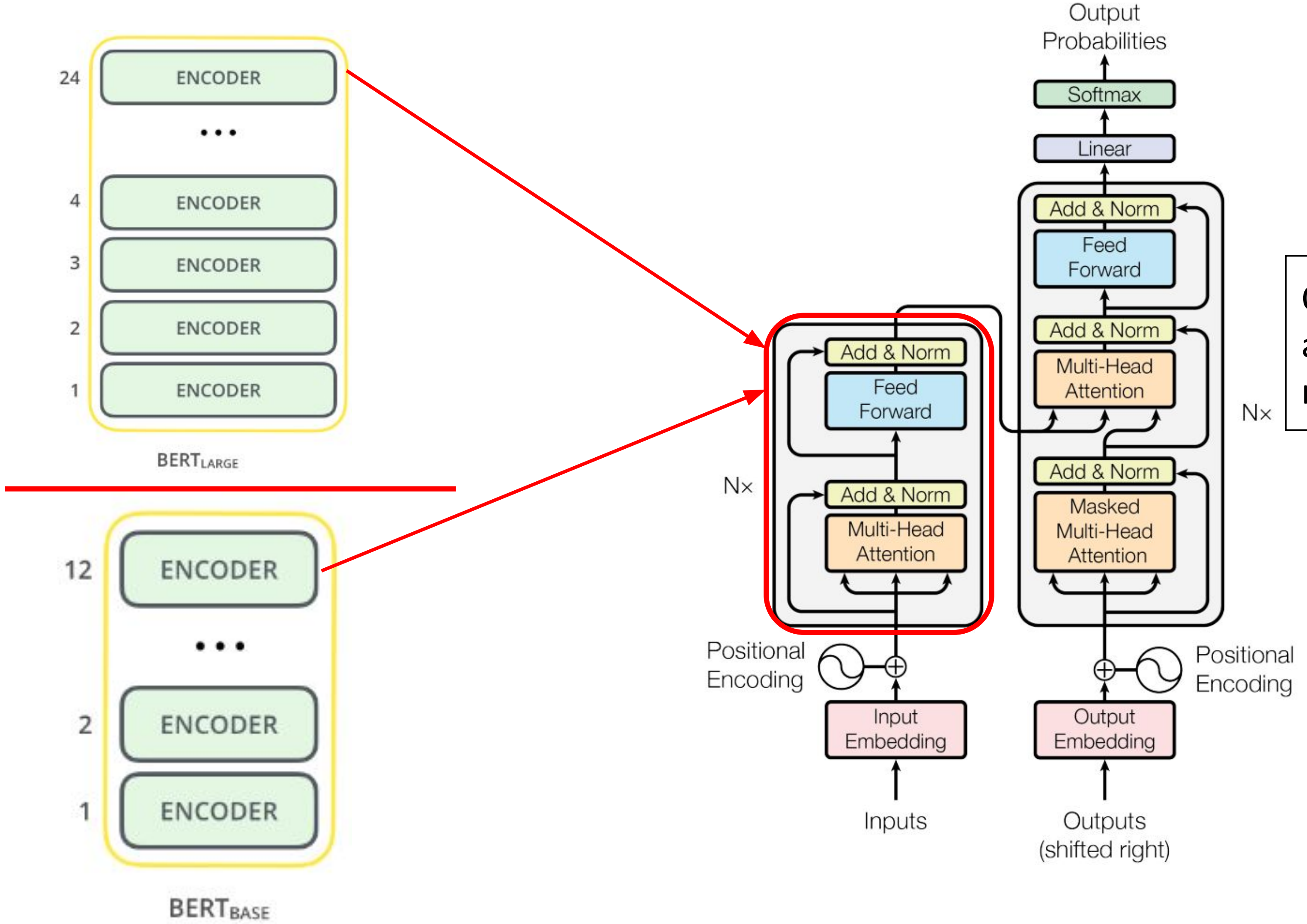


Image source: Vaswani et. al. (2017)

The Transformers Architecture



Q: How does textual information flow from a set of tokens to mathematical representation via BERT?

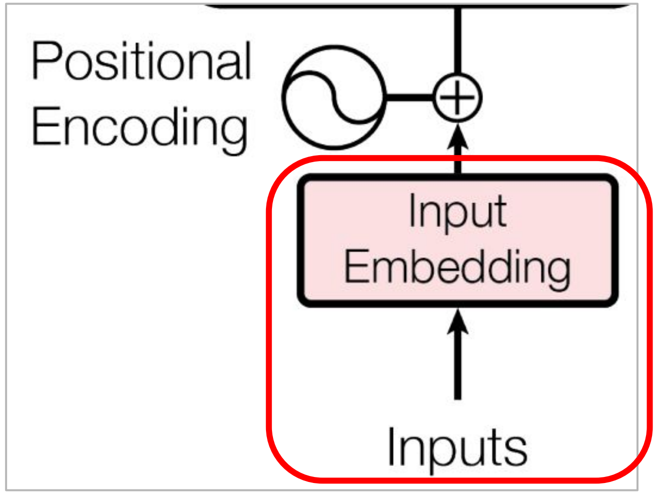
Input Embeddings

Input Sentence: *“ Hello, how are you?”*

Tokenization: *“ Hello, how are you?” → [“Hello”, “,” , “how”, “are”, “you”, “?”]*

Numericalization:

[“Hello”, “, “, “how”, “are”, “you”, “?”] → [34, 90, 15, 684, 55, 193]



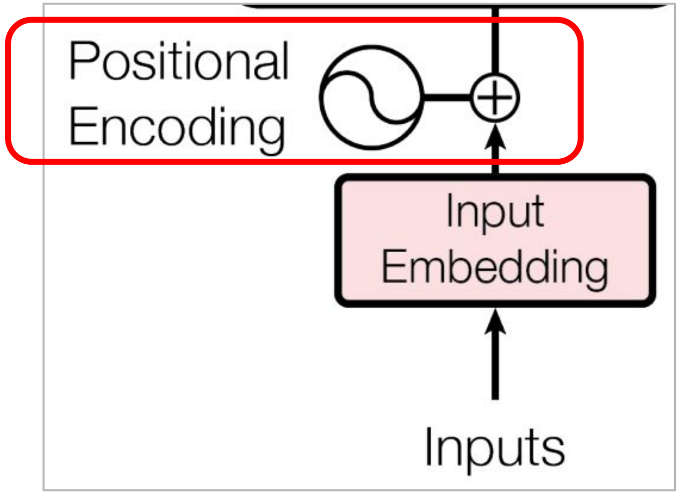
Padding:

[“<pad>”, “<pad>”, “<pad>”, “Hello”, “, “, “how”, “are”, “you”, “?”] → [5, 5, 5, 34, 90, 15, 684, 55, 193]

if the *input_length* was set to 9.

Positional Encoding

- As of yet, the model contains no recurrence and no convolution
  - in order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position of the tokens in the sequence
  - Add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks



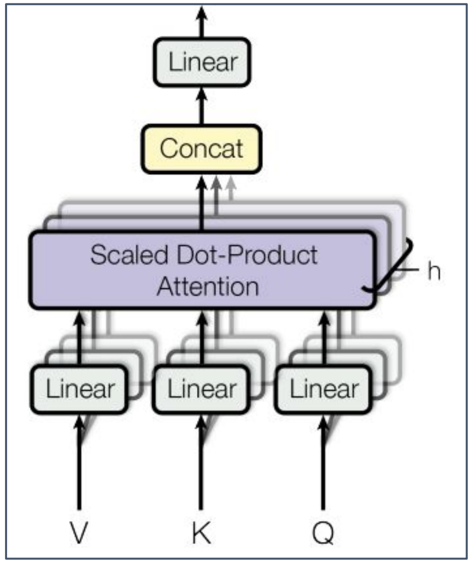
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

For deep dive: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Attention!

The Why

- Lower Computational Complexity.
- Computation of self-attention can be parallelized.
- Path length between long-range dependencies is shorter via self-attention.



$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

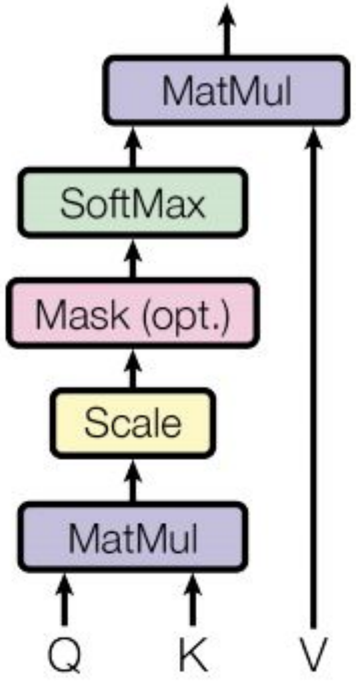
- Multi-head attention concatenates the dot-product attention computed for each attention head.
- Each attention head is computed based on learnable parameters Q, K, and V; which are also placeholders for different input matrices.
- For each input token, use its query vector (Q) to score against all the other key vectors (K)
- Sum up the value vectors (V) after multiplying them by their associated scores.

Masking: A simulated learning objective

The training objective for BERT-like language models relies on “predicting the masked word”.

While computing self-attention, the learnable parameters are computed based on how closely was the masked word predicted.

Before providing input, BERT tokenization allows one to mask a certain %age of words from the input set of sentences.



Other Architectures

RoBERTa *vs.* BERT vs. DistilBERT

- In BERT, masking is performed only once at data preparation time, and they basically take each sentence and mask it in 10 different ways.
  - At training time, the model will only see those 10 variations of each sentence.

- On the other hand, **in RoBERTa, the masking is done while training**.
  - Each time a sentence is incorporated in a minibatch, **it gets its masking done dynamically**.
  - The number of potentially different masked versions of each sentence is not bounded like in BERT.

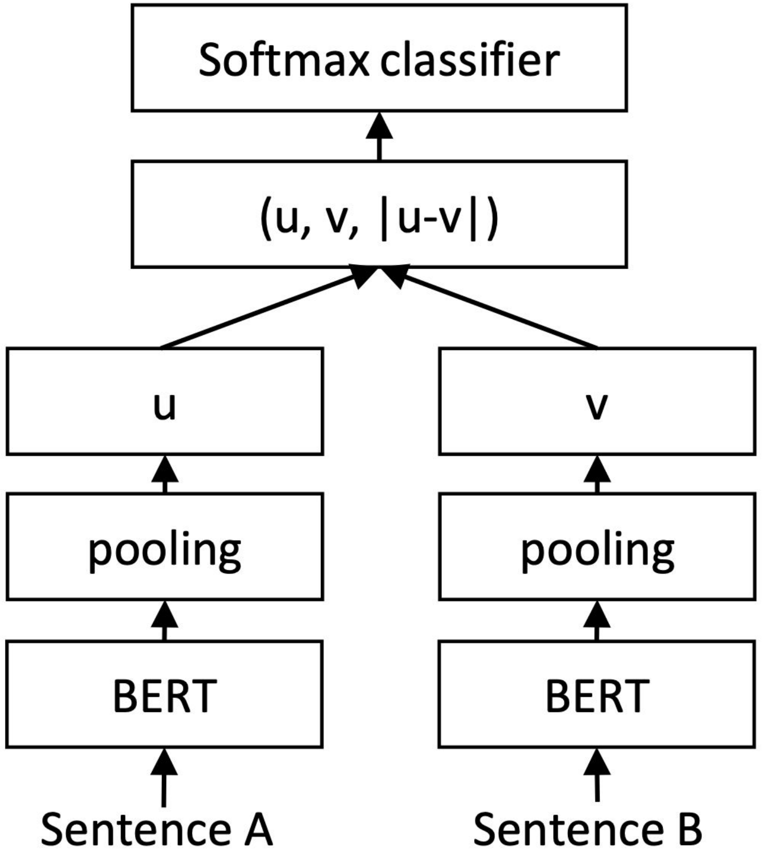
	BERT	RoBERTa	DistilBERT
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base**: 66
Training Time	**Base**: 8 x V100 x 12 days* **Large**: 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large**: 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation

Sentence-BERT Architecture

- Sentence-BERT introduces pooling to the token embeddings generated by BERT to create a fixed sentence embedding.
  - When this network is fine-tuned on Natural Language Inference (NLI) data it does become apparent that it is able to encode the semantics of sentences.

- These can be used for unsupervised tasks (*e.g.*, semantic textual similarity) or classification problems where they achieve state-of-the-art results.

- SBERT is also computationally more efficient as compared to BERT.



GPT-n Architecture

- Autoregressive models are pretrained on the classic language modeling task.
  - Guess the next token having read all the previous ones.
- They correspond to the decoder of the original transformer model, and a mask is used on top of the full sentence so that the attention heads can only see what was before in the text, and not what's after.
- Although those models can be fine-tuned and achieve great results on many tasks, the most natural application is text generation. A typical example of such models is GPT.
- The key difference: <u>No encoder block</u>

GPT-n: Use Cases

- The simplest way to run a trained GPT-2 is to allow it to ramble on its own (which is technically called generating unconditional samples).
- Alternatively, we can give it a prompt to have it speak about a certain topic (*i.e.,* generating interactive conditional samples).
- In the rambling case, we can simply hand it the start token and have it start generating words.
- The trained model uses <|endoftext|> as its start token.

Transfer Learning: Examples

(w/ some ongoing investigations)

Fine-Tuning for NLP Tasks: Transfer Learn

- The main benefit behind Transformers is that once pre-trained, Transformers can be fine-tuned for numerous downstream tasks and often perform really well out of the box.
- This is primarily due to the fact that the Transformer already **'understands'** context for a word which allows training to focus on learning how to do
  - Question Answering
  - Language Generation
  - Named Entity Recognition
  - …
  - *Anything which utilizes features from text/language to perform a classification or regression or generation task.*

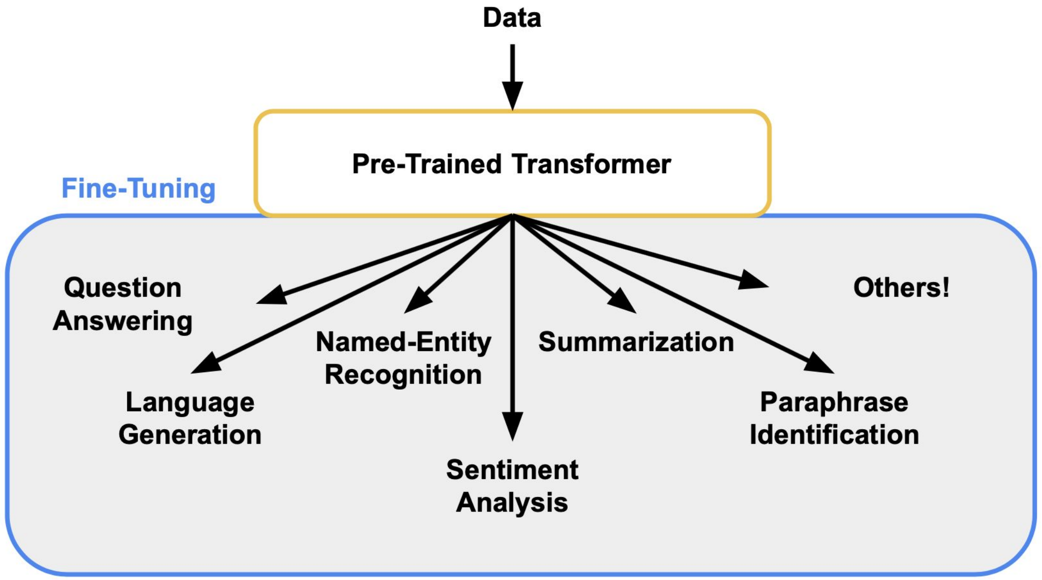


Image Source: https://www.assemblyai.com/blog/fine-tuning-transformers-for-nlp/

Neural Machine Translation (NMT)

- NMT enables the use of neural architecture to translate text from one natural language to another.
- Statistical Machine Translation (SMT) performance was surpassed using Transformers architecture [BERT (Vaswani et. al., 2017)]
- Winner, SMT competition at ICON 2014 (Prabhugaonkar et. al., 2014)
  - Task of translating *from English, Bengali, Marathi, Tamil, and Telugu* to Hindi.
  - Use of Hierarchical Phrase-based SMT decoder with KenLM (language model).
- Arrival of NMT using recurrent architectures. (Bahdanu et. al., 2014; Sutsekever et. al., 2014; Luong et. al., 2015)
- State-of-the-art (SoTA) achieved using (massive) Multilingual NMT systems.
  - Based on Transformers architecture. (Aharoni et. al., 2019; Costa-jussà et. al., 2022)
  - Quoted in Sky News article on Facebook's NLLB system on Evaluation using BLEU[1]

[1]Meta claims breakthrough in 'superpower' AI translation as it interprets more than 200 languages | Science & Tech News | Sky News

NMT still imperfect? – Automatic Post Editing

- Automatic Post Editing is the task of correcting machine translated output using various methods.
  - Statistical methods (Chatterjee et al., WMT 2015; Libovický et. al., 2016)
  - Neural methods (Chatterjee et al., 2018; Chatterjee et al., WMT 2020)

- Requires human post-editors to build post-editing resource by correcting translation output manually.

- Automatic Post Editing Shared Task Organization
  - Introduced English-Marathi resource in 2022 edition.
  - Introducing English-Hindi resource in 2023 edition.

How do you assess Translation Quality automatically? - Quality Estimation

- Quality Estimation is the task for automatically assessing the quality of translated output using various methods.
    - Statistical methods / Machine Learning (Kozlova et. al., 2016)
    - Deep Neural Networks (Ranasinghe et. al., 2020) [Current SoTA]

- Requires (at least 3) human translators to build a resource where they assess the quality manually to generate z-score.

- Based on normalized z-score, it is a regression task to judge translation quality using any methods stated above.

- Quality Estimation Shared Task Organization
    - Introduced English-Marathi resource in 2022 edition.
    - Introducing English-Hindi resource in 2023 edition.
    - Introducing English-Sinhala resource in 2023/2024 edition.

Thank you!

Questions?

References

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the association for computational linguistics, 5, pp.135-146.

Conneau, A., Lample, G., Ranzato, M.A., Denoyer, L. and Jégou, H., 2017. Word translation without parallel data. International Conference on Learning Representations. 2018.

Artetxe, M., Labaka, G. and Agirre, E., 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. Association for Computational Linguistics (Volume 1: Long Papers), pages 789–798

Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Conneau, A. and Lample, G., 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, *32*.

Artetxe, M., Ruder, S. and Yogatama, D., 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.

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

Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

Luong, M.T., Le, Q.V., Sutskever, I., Vinyals, O. and Kaiser, L., 2015. Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114.

Aharoni, R., Johnson, M. and Firat, O., 2019. Massively multilingual neural machine translation. arXiv preprint arXiv:1903.00089.

NLLB Team, Marta R. Costa-jussà , James Cross , Onur Çelebi , Maha Elbayad , Kenneth Heafield , Kevin Heffernan , Elahe Kalbassi , Janice Lam , Daniel Licht , Jean Maillard , Anna Sun , Skyler Wang Guillaume Wenzek , Al Youngblood et. al. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. arXiv preprint arXiv:2207.04672.

Prabhugaonkar, N.R., Pawar, J.D., Nagvenkar, A.S., Bhattacharyya, P., Kanojia, D. and Shrivastava, M., 2014. PanchBhoota: Hierarchical phrase based machine translation systems for five Indian languages.

Rajen Chatterjee, Marco Turchi, and Matteo Negri. 2015. The FBK Participation in the WMT15 Automatic Post-editing Shared Task. In Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 210–215, Lisbon, Portugal. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 646–654, Berlin, Germany. Association for Computational Linguistics.

Findings of the WMT 2018 Shared Task on Automatic Post-Editing (Chatterjee et al., 2018)

Findings of the WMT 2020 Shared Task on Automatic Post-Editing (Chatterjee et al., WMT 2020)

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. YSDA Participation in the WMT'16 Quality Estimation Shared Task. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 793–799, Berlin, Germany. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-Level Direct Assessment. In Proceedings of the Fifth Conference on Machine Translation, pages 1049–1055, Online. Association for Computational Linguistics.

UNIVERSITY OF
SURREY