Natural Language Processing

Parsa Bagherzadeh

Jan 8, 2024

NLP

- Interaction between computers and human languages
- Enabling machines to understand, interpret, and generate human language
- Using various methods:
  - Computational linguistics
  - Machine learning
  - Semantic theory
  - ...
- Process and analyze large amount of unstructured textual data

Applications (in medicine)

- Information extraction
  - Drug mentions
  - Treatment modalities
- Clinical Trial Matching
  - Matching patient to proper trial based on medical history
- Patient Risk Stratification
- Medical Coding
  - Assigning proper codes to diagnostic and procedure services
- Pharmacovigilance and Adverse Event Monitoring
- Automated Report Generation

Common tasks

- Document classification: Classifying notes for medical speciality
- Named-entity recognition: Classifying entities into pre-determined categories

  Chest CT was negative for pulmonary embolism

- Co-reference resolution: Finding all mentions that refer to the same entity

  The patient was found to have chronic obstructive pulmonary disease. The patient continued to become weaker and it was clear that her COPD was end-stage.

- Relation extraction: Detecting semantic relationships between entities

  Finding

  A CT was performed which reveled the presence of pseudocyst in the pancreas

Tokenization

- The first step in most NLP pipelines
- Breaking a text into smaller units (tokens)
- Each token is a meaningful unit

Text:	*Chest CT was negative for pulmonary embolism*
Tokens:	*Chest*, *CT*, *was*, *negative*, *for*, *pulmonary*, *embolism*

- Tokenization is not as simple as breaking at the spaces
- Hyphenated terms, comma, parenthesis

Example: *The patient presented with gastro-esophageal reflux disease (GERD) and was prescribed proton-pump inhibitors*

Part-of-Speech tagging

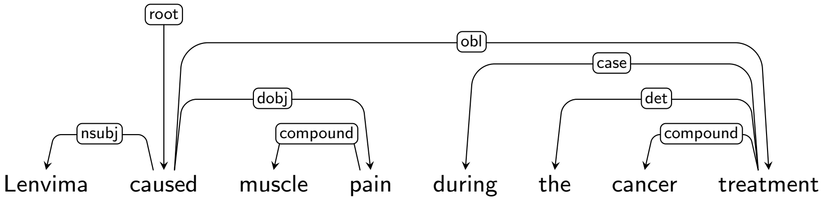
- The syntactic category of a word
- Assigned based on the contribution of words to the meaning of the phrase
- Common POS: Nouns, verbs, adjectives, pronouns, prepositions, ...

Chest	*CT*	*was*	*negative*	*for*	*pulmonary*	*embolism*
NN	NN	VBD	JJ	IN	JJ	NN

- What is the use?
- Named-entities are nouns/noun phrases!

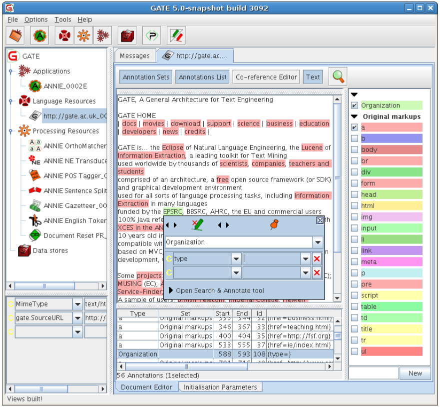
Dependency parse

- Syntactic structure consists of relations between words
- Asymmetric between a head (governor) and a dependent
- The relations are typed and called dependency



Some NLP tools

- GATE
  - Java-based
  - Modular pipeline
  - Visulization of annotations
- Spacy
- Stanza
  - Python-based



GATE GUI

Neural NLP

- The dominant approach in the past decade (deep learning)
- The tasks are often modeled as classification tasks
- A neural model is trained end-to-end
- Tokens are represented in vector spaces

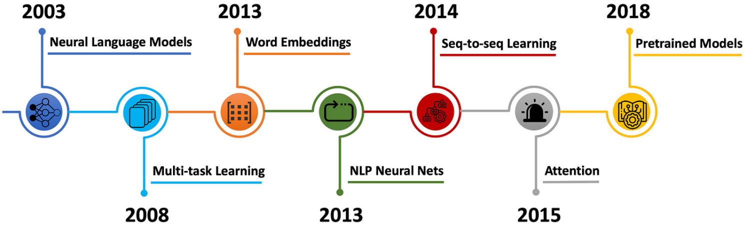


Figure from Medium

Word embeddings

- Each term w_t is represented by vector x_t
- Related terms are often closer to each other (under certain conditions)
- Can be used as the inputs to a neural model
- **Examples:** Word2Vec, GLoVE

  (available for download in .txt format)

$$
\begin{array}{rcl|ccccccc|}
\hline
\text{this} & \rightarrow & 0 & 0.9 & -0.5 & 0.80 & \ldots & -0.2 & -0.3 & -0.5 \\
\text{he} & \rightarrow & 1 & -0.7 & 0.0 & 0.5 & \ldots & 0.2 & 0.6 & 0.3 \\
\text{CT} & \rightarrow & 2 & -0.6 & -1.1 & -0.2 & \ldots & -2.2 & 1.2 & -0.1 \\
\text{cancer} & \rightarrow & 3 & -0.0 & -0.5 & 0.0 & \ldots & -0.2 & 0.5 & -1.6 \\
\text{analysis} & \rightarrow & 4 & 0.0 & 0.7 & 0.8 & \ldots & 1.0 & -0.1 & 2.0 \\
\vdots & & \vdots & & & & \vdots & & & \\
\text{tumor} & \rightarrow & |V| & -0.6 & -0.8 & -1.6 & 0.2 & -1.2 & -1.5 & -2.3 \\
\hline
\end{array}
$$

CNN for text classification

- Textual input is seen as a matrix (just like an image)
- Convolutional filters are applied to it
- The filter size however needs to be the same as embedding size (d)

$$c_i = f(X_{i:i-l+1} * W + b) \qquad W \in \mathbb{R}^{l \times d}$$

- The result is a feature map:

$$C = [c_1, c_2, \dots]$$

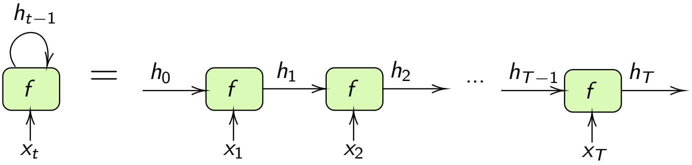
$$\begin{array}{r|c|c|c|c|c|} \hline \textit{Labetalol} & 7 & 3 & 0 & 9 & 1 \\ \hline \textit{is} & 3 & 5 & 1 & 6 & 0 \\ \hline \textit{not} & 4 & 5 & 1 & 0 & 0 \\ \hline \textit{effective} & 0 & 0 & 1 & 1 & 2 \\ \hline \end{array} \; * \; \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 \\ \hline 2 & 0 & 2 & 1 & 0 \\ \hline \end{array} \; = \; \begin{array}{|c|} \hline 30 \\ \hline 19 \\ \hline 8 \\ \hline \end{array}$$

$$X \qquad\qquad W \qquad\qquad C$$

Recurrent Neural Nets

- A recurrent layer can process sequences $\langle x_1, x_2, \ldots, x_T \rangle$
- It provides the hidden representations $\langle h_1, h_2, \ldots, h_T \rangle$

$$h_t = f(x_t, h_{t-1}) = tanh(x_t W + h_{t-1} \tilde{W}) \quad W \in \mathbb{R}^{d_{in} \times d_h} \quad \tilde{W} \in \mathbb{R}^{d_h \times d_h}$$



- h_t is a representation of the sequence up to position t
- h_T (last hidden representation) can be used for document classification

Context

- How a word (its meaning) should be represented?

" You shall know a word by the company it keeps"

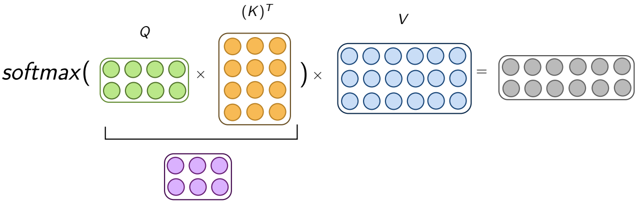
John R. Firth

Lenvima is effective for the treatment of thyroid cancer

Dot product Attention: A general form

- Mapping a query (Q) and a set of key-value (K,V) pairs to an output
- The output is computed as a weighted sum of the values
- The weight for each value is computed by the dot product of the query with the corresponding key

$$Attention(Q, K, V) = softmax(\frac{Q(K)^T}{\sqrt{d_h}})V \tag{1}$$

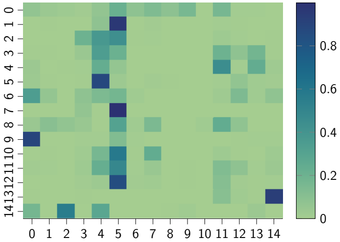


Self-attention

- If $Q = K = V = X$ then we have self-attention
- Obtains contextualized representation for each token
- Each token is represented using other tokens (and itself)

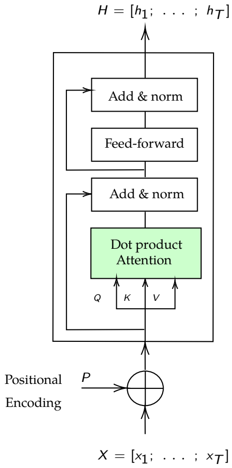
$$Attention(X, X, X) = softmax(\frac{(XW^q)(XW^k)^T}{\sqrt{d_h}})XW^v \tag{2}$$

- Trainable parameters of the model
- Avoiding trivial attention weights
- Query and Key can have different sizes
- What if the context is not important?



Transformers (encoder)

- RNNs process a sequence one token at a time
- They are not bi-directional
- Transformers process all tokens simultaneously
- Positional encoding captures the order
- Self-attention gives contextualized representations
- Residual connections to avoid exploding gradients
- FF layer for more representation learning

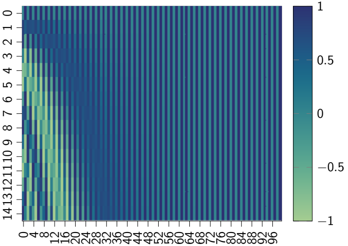


Positional Encoding

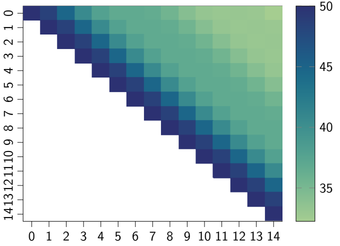
- Captures the absolute positions
- Randomly initialized encoding (learned)

  or
- Sinusoidal encoding (constant)

$$P_{t,2i} = sin(t/10000^{(2i/d)})$$
$$P_{t,2i+1} = cos(t/10000^{(2i/d)})$$



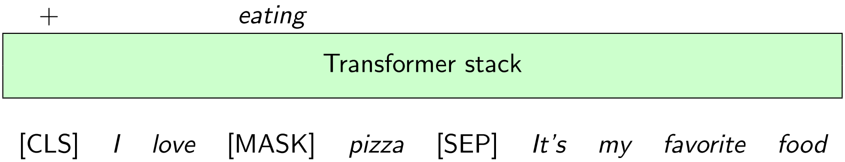
- Adjacent tokens have similar encodings
  - Large dot-product/cosine similarity



Dot product

BERT 

- A Transformer-based language model
- Comprises at least 12 stacked Transformers
- Unsupervised pre-training:
  - Masked language modeling (MLM)
  - Next sentence prediction (NSP)
- Pre-trained on WikiPedia and BookCorpus



BERT embedding layer

- BERT embedding layer sums three types of embeddings:
  - Token embedding
  - Positional encoding
  - Segment embedding (representing the span of sentences)

$$
\begin{array}{cccccccccc}
S_A & S_A & S_A & S_A & S_A & S_A & S_B & S_B & S_B & S_B \\
+ & + & + & + & + & + & + & + & + & + \\
P_{1,:} & P_{2,:} & P_{3,:} & P_{4,:} & P_{5,:} & P_{6,:} & P_{7,:} & P_{8,:} & P_{9,:} & P_{10,:} \\
+ & + & + & + & + & + & + & + & + & + \\
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\
\hline
\text{[CLS]} & \textit{I} & \textit{love} & \text{[MASK]} & \textit{pizza} & \text{[SEP]} & \textit{It's} & \textit{my} & \textit{favorite} & \textit{food}
\end{array}
$$

- Not all Transformer-based models have segment embedding
- RoBERTa does not have a NSP task

Vision Transformer

- An image is seen as a sequence of patches
- Positional encoding capture the relative position of patches
- Compared to CNNs, it better captures the spatial features