Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Evaluation Exercise II

Borchert, Konak, Dr. Schapranow

Data Management for Digital Health

Winter 2020

Exercise II
Topics

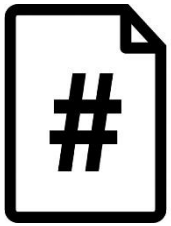
- Medical Use Case Oncology
- Text Data & NLP

Exercise II
Key Stats

Hasso Plattner Institut

25 Questions	Students	Average score	Average time
50 Points	48 Passed	41.2 / 82 %	91.5 min

<< 3h



Partly correct

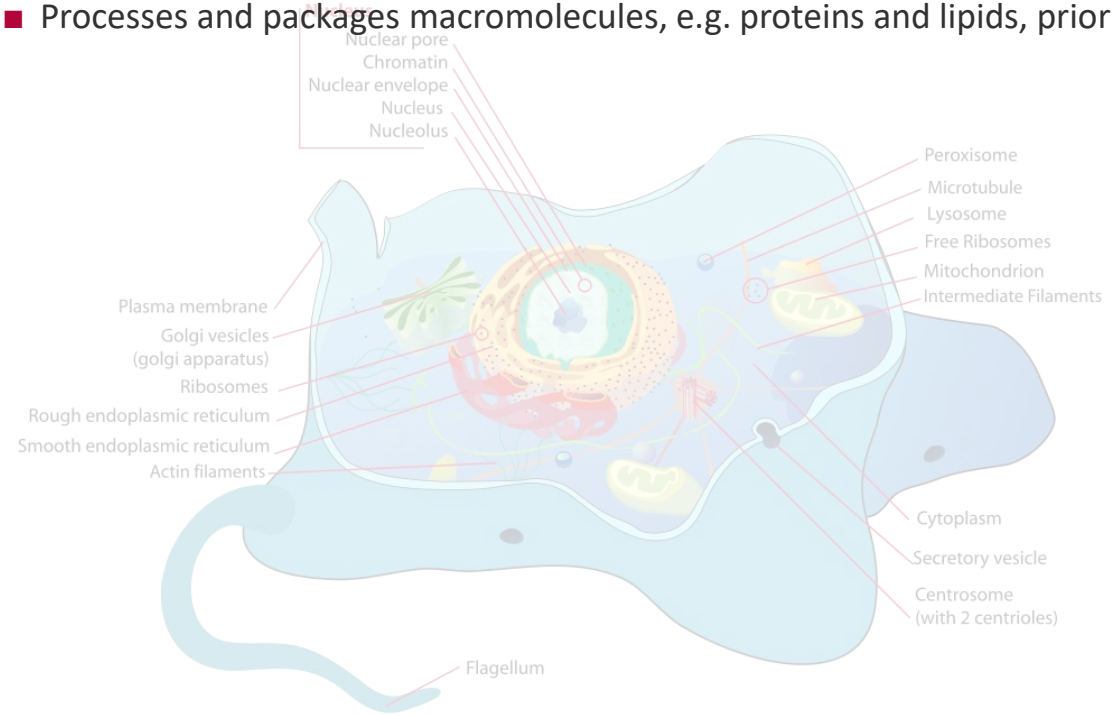
Wrong

Q8. Which of the following cell organelles plays an important for the protein biosynthesis?

- Golgi apparatus.
- Ribosomes.
- mDNA.
- DNA polymerase.

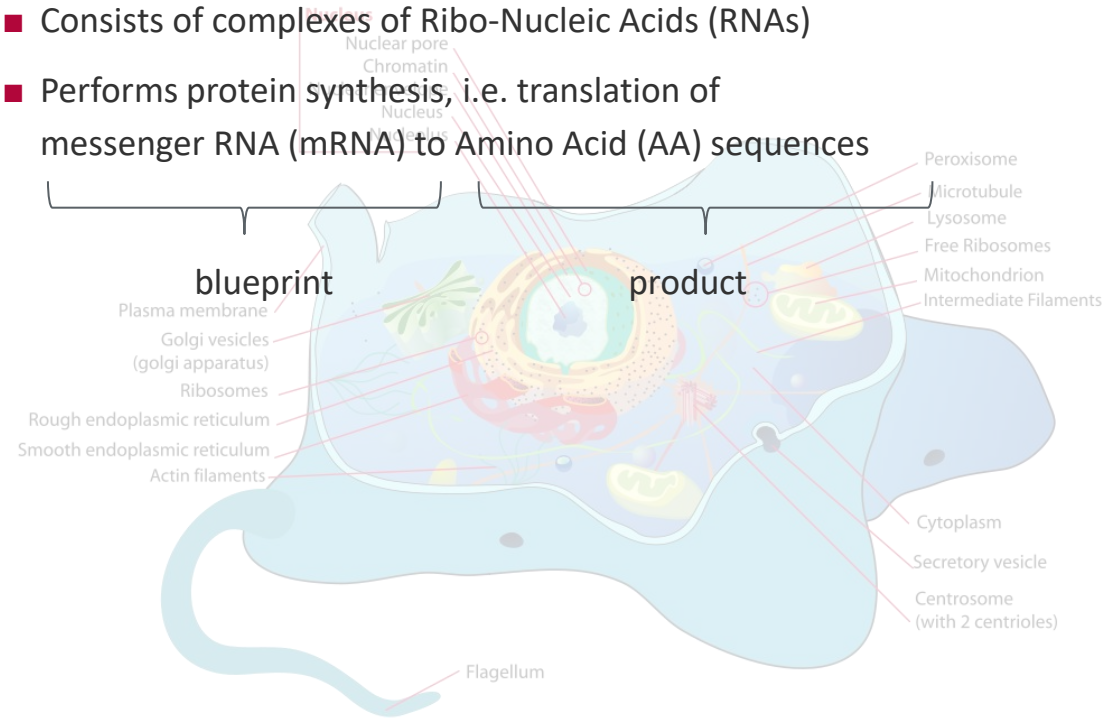
Components of Eukaryotic Cells (Organelles)
Golgi Apparatus

- Processes and packages macromolecules, e.g. proteins and lipids, prior to transport



Components of Eukaryotic Cells (Organelles)
Ribosome

- Consists of complexes of Ribo-Nucleic Acids (RNAs)
- Performs protein synthesis, i.e. translation of messenger RNA (mRNA) to Amino Acid (AA) sequences



What to take Home?

- Mitochondria: Power supply for the cell

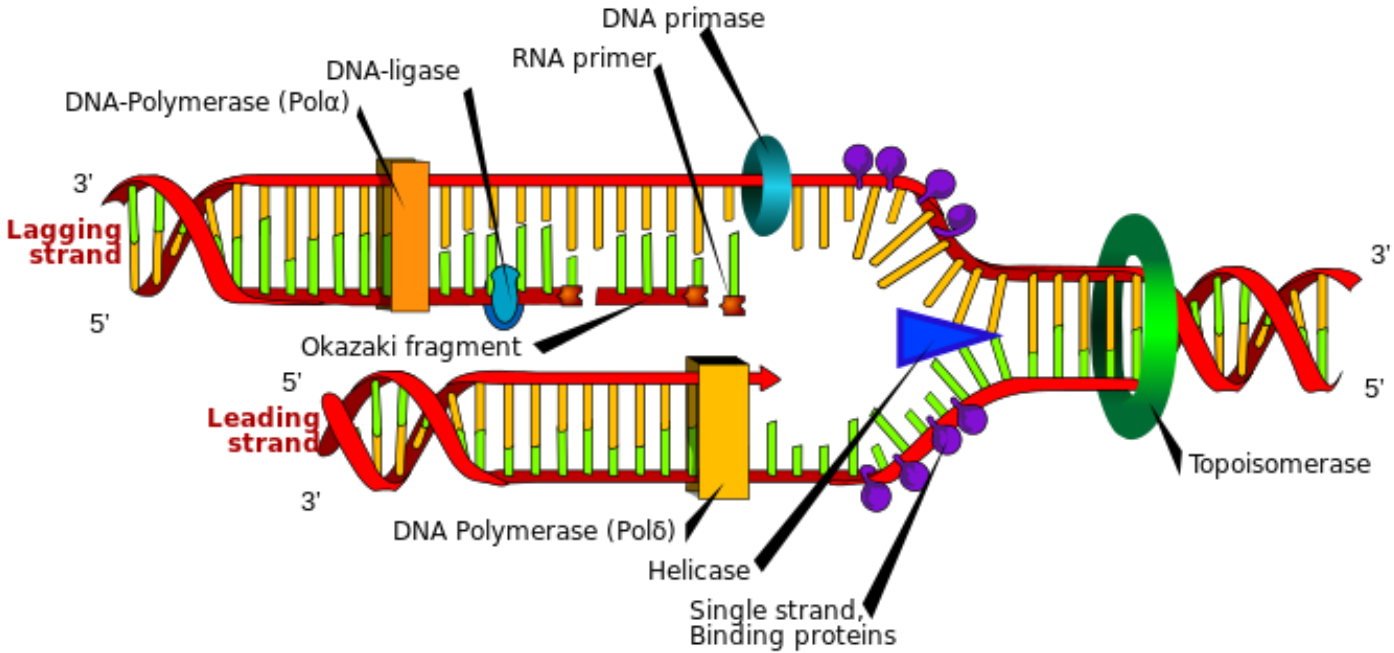
- Cell core: Contains source code, i.e. DNA

- Endoplasmic reticulum: Provides transport network

- Ribosomes: Compiler, e.g. mRNA to AA

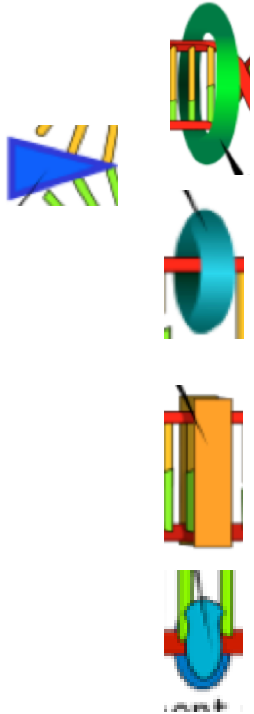
- Golgi apparatus: Packaging, e.g. proteins

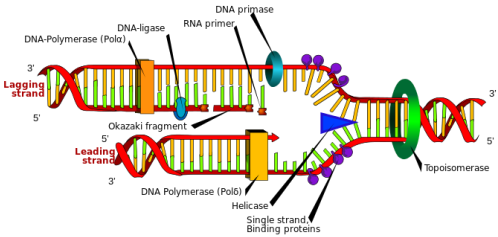
DNA Replication during Synthesis Phase



DNA Replication during Synthesis Phase

1. Initiation
   - Topoisomerase unwinds/overwinds DNA
   - Helicase unzips DNA at specific origins
   - Primase adds primer for binding of polymerase
2. Elongation
   - DNA polymerase
     - Extends DNA only in 5’ → 3’ direction using a template strand
     - Performs proofreading of replicated strand
   - DNA ligase seals strand breaks
3. Termination: Replication comes to an end





Q8. Which of the following cell organelles plays an important for the protein biosynthesis?

- Golgi apparatus. 
- Ribosomes. 
- mDNA. 
- DNA polymerase. 

Q15. Your goal is to extract all mentioned of drug names from clinical notes and map all mentions of the same drug to their canonical identifier, e.g. the Anatomical Therapeutic Chemical (ATC) code. Please select all appropriate NLP tasks that apply in this scenario.

Hasso Plattner Institut

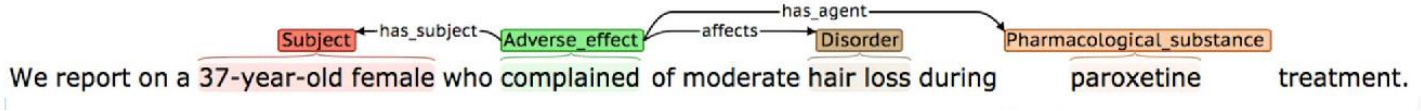
- Sentiment Analysis.
- Named Entity Recognition.
- Named Entity Normalization.
- Information Retrieval.

Some Common NLP Tasks

Level: sentences / paragraphs

- Sentiment Analysis: „I feel a bit sad“ → Affective State: Polarity: ☹ Strength: 0.4
- Dialogue Systems: „I feel a bit sad“ → „Why do you feel sad?“

Tasks for Information Extraction in Clinical Context



Named Entity Recognition

Identifying instances of classes, e.g., *paroxetine* as a pharmacological substance

Named Entity Normalization

(aka Entity Linking)

Mapping of entities to unique identity, e.g., *hair loss* to its ICD-10 code

(Temporal) Relation Extraction

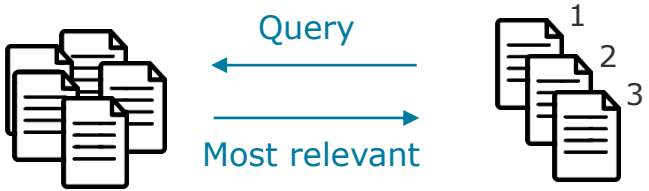
Identifying *n*-ary relations between entities, e.g., *adverse_effect* (*paroxetine, hairloss*)

Thompson, Paul, et al. "Annotation and detection of drug effects in text for pharmacovigilance." *Journal of cheminformatics* 10.1 (2018): 37.

Some Common NLP Tasks

Level: collections of text documents

- Information Retrieval



- Topic modeling

Q15. Your goal is to extract all mentioned of drug names from clinical notes and map all mentions of the same drug to their canonical identifier, e.g. the Anatomical Therapeutic Chemical (ATC) code. Please select all appropriate NLP tasks that apply in this scenario.

- Sentiment Analysis. 
- Named Entity Recognition. 
- Named Entity Normalization. 
- Information Retrieval. 

Q16. *Lemmatization* is a common preprocessing step in NLP pipelines. It transforms inflected tokens to their base form, incorporating the context and meaning of the word, as opposed to simple stemming, […]

Hasso Plattner Institut

[…] which performs a similar operation but does not respect context and meaning. Let us consider the tokens “drug”, “druggable”, “drugged” or “drugs”, which would all be reduced to their base form “drug”. On which of the following linguistic levels is lemmatization operating on?

- Syntax.
- Phonology.
- Morphology.
- Phonetics.

Aspects of Human Language Studied by Linguists

(Sample of) structural subfields of linguistics:

- Phonetics := study of sounds of human language
- Phonology := study of sound systems in human language
- Morphology := study of the formation and internal structure of words
- Syntax := study of the formation and internal structure of sentences
- Semantics := study of the meaning of sentences
- Pragmatics := study of the way sentences with their semantic meaning are used for particular communicative goals

Bender, Emily (2013): Linguistic Fundamentals for Natural Language Processing

Q16. *Lemmatization* is a common preprocessing step in NLP pipelines. It transforms inflected tokens to their base form, incorporating the context and meaning of the word, as opposed to simple stemming, […]

Hasso Plattner Institut

[…] which performs a similar operation but does not respect context and meaning. Let us consider the tokens “drug”, “druggable”, “drugged” or “drugs”, which would all be reduced to their base form “drug”. On which of the following linguistic levels is lemmatization operating on?

Note: Correct lemma depends on part-of-speech, e.g.

saw (verb) -> see
saw (noun) -> saw

→ Indistinguishable without syntactic context

- Syntax. ✓
- Phonology. ✗
- Morphology. ✓
- Phonetics. ✗

Q17. Extraction of accurate phenotypes from clinical notes is an important application of NLP as discussed in class. Why is it not sufficient to use International Code of Disease (ICD) codes in the EHR for this purpose?

- ICD codes are used for billing purposes and may therefore not serve the purpose of our downstream application.
- ICD codes are not machine-readable and cannot easily be turned into features for ML algorithms.
- There are too many ICD codes to efficiently store them in a relational database.
- ICD codes fall under higher levels of data protection than clinical free-text, but cannot be de-identified.

Discharge Summary

Provider: Ken Cure, MD

Patient: Patient H Sample **Provider's Pt ID:** 6910828 **Sex:** Female

Attachment Control Number: XA728302

HOSPITAL DISCHARGE DX

- ☐ 174.8 Malignant neoplasm of female breast: Other specified sites of female breast
- ☐ 163.8 Other specified sites of pleura.

HOSPITAL DISCHARGE PROCEDURES

1. 32650 Thoracoscopy with chest tube placement and pleurodesis.

HISTORY OF PRESENT ILLNESS

The patient is a very pleasant, 70-year-old female with a history of breast cancer that was originally diagnosed in the early 70's. At that time she had a radical mastectomy with postoperative radiotherapy. In the mid 70's she developed a chest wall recurrence and was treated with further radiation therapy. She then went without evidence of disease for many years until the late 80's when she developed bone metastases with involvement of her sacroiliac joint, right trochanter, and left sacral area. She was started on Tamoxifen at that point in time and has done well until recently when she developed shortness of breath and was found to have a larger pleural effusion. This has been tapped on

Q17. Extraction of accurate phenotypes from clinical notes is an important application of NLP as discussed in class. Why is it not sufficient to use International Code of Disease (ICD) codes in the EHR for this purpose?

- ICD codes are used for billing purposes and may therefore not serve the purpose of our downstream application. 
- ICD codes are not machine-readable and cannot easily be turned into features for ML algorithms. 
- There are too many ICD codes to efficiently store them in a relational database. 
- ICD codes fall under higher levels of data protection than clinical free-text, but cannot be de-identified. 

Q23. Which are examples of useful features to represent tokens for an ML algorithm that classifies tokens as I/O/B (In/Out/Beginning) for named entity recognition?

Hasso Plattner Institut

- Boolean feature indicating a dictionary match.
- Number of uppercase characters within the token.
- Memory address of the string in RAM.
- Boolean feature indicating whether the token left of the current token has the part-of-speech-tag determiner.

ML-based Approaches for Named-Entity Recognition

- ML-based NLP requires training corpus (= dataset in NLP)

BRAF V600E is a driver mutation found in multiple tumor types

B I O O O O O O O O O

- Training data := Sequences of tokens $\mathbf{x} = (x_1, ..., x_n)$, e.g. $x_1 =$ "BRAF", $x_6 =$ "mutation"
- Labels := Sequences of I/O/B tags $\mathbf{y} = (y_1, ..., y_n)$, e.g. $x_1 =$ "B", $y_6 =$ "O"
- Simple approach to ML-based NER
  - Turn token into feature vector v
  - Classify each token independently as I/O/B
    - $f(v(\text{"BRAF"})) = B$
    - $f(v(\text{"V600E"})) = I$
    - $f(v(\text{"mutation"})) = O$

Features: Word Identity

- Simple feature: Identity of current token
- One-hot-encoding : create a feature vector with length equal to size of vocabulary and set 1 if word == ith element else 0
- Vocabulary size is typically restricted to most common *k* tokens
- Special feature for unknown tokens (unknown tokens are often named entities!)
- Resulting feature vectors are very sparse (almost all elements are 0)

$$v(\text{“apple”}) \rightarrow \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad v(\text{“cancer”}) \rightarrow \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

Word Shape Features

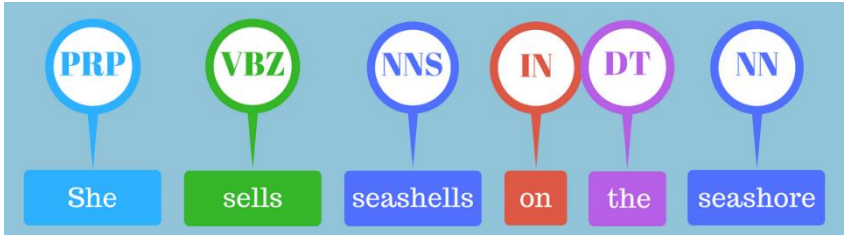
- Word shape features := abstract letter patterns
- Prefixes
- Suffixes (e.g., word ends with *"–itis"* → disease)
- character n-grams

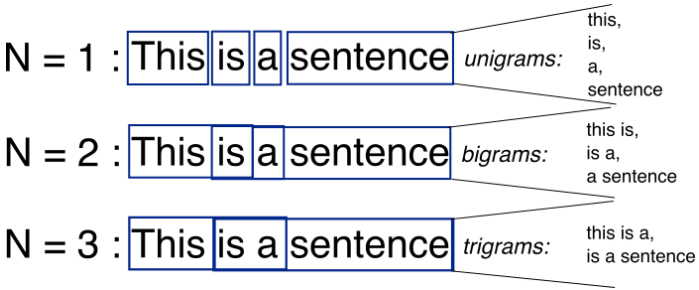
Feature	*About feature*	*Regular Expression*
Initcaps	First letter is in uppercase	[A-Z]\w+
Initcapsalpha	First letter is in uppercase. Second letter is in lowercase.	[A-Z][a-z]\w+
All caps	All letters are in uppercase	[A-Z]+
Caps mix	Mixture of uppercase and lowercase letters	[A-Za-z]+
Has digit	Protein name has a number in the middle	\w+[0-9]\w+
Single digit	Ranges from 0-9	[0-9]
Double digit	Two digit numbers	[0-9][0-9]
Natural numbers	Any natural number	[0-9]+
Real numbers	Decimal numbers /numbers with comma	[-0-9]+[.,]+[0-9.,]+
Has dash	Protein name with dash in middle	\w+ - \w+
Init dash	First character is a dash	- \w+
End dash	Last character is a dash	\w+ -
Alphanumeric (starts with alphabet)	Combination of alphabets and numbers. First character is an alphabet.	\w+ [A-Za-z] \w+ [0-9] \w+
Alphanumeric (starts with number)	Combination of alphabets and numbers. First character is a number.	\w+ [0-9] \w+ [A-Za-z] \w+
Roman letter	Any roman letter	[IVXDLCM]+
Has Roman	Any roman letter in the middle	\w+ [IVXDLCM]+ \w+
Greek	Any Greek letter	\w+ [aßYÖE]
Has Greek	Any Greek letter in middle	\w+ [aßYÖE] \w+
Punctuation	Protein name with punctuation	

Raja, Kalpana, Suresh Subramani, and Jeyakumar Natarajan. "A hybrid named entity tagger for tagging human proteins/genes." *International journal of data mining and bioinformatics* 10.3 (2014): 315-328.

More Token Features

- Dictionary match
- Part-of-speech tags (e.g., is the word a proper noun)
- Context features
  - Distributional semantics: “a word is characterized by the company it keeps” (J.R. Firth, 1957)
    ➔ Surrounding tokens can be more important than token itself
  - word n grams (common: trigrams with current, left and right word)





https://medium.com/analytics-vidhya/pos-tagging-using-conditional-random-fields-92077e5eaa31
https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram

Q23. Which are examples of useful features to represent tokens for an ML algorithm that classifies tokens as I/O/B (In/Out/Beginning) for named entity recognition?

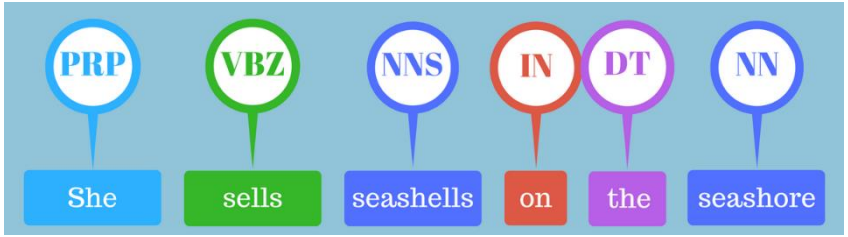
- Boolean feature indicating a dictionary match. 
- Number of uppercase characters within the token. 
- Memory address of the string in RAM. 
- Boolean feature indicating whether the token left of the current token has the part-of-speech-tag determiner. 

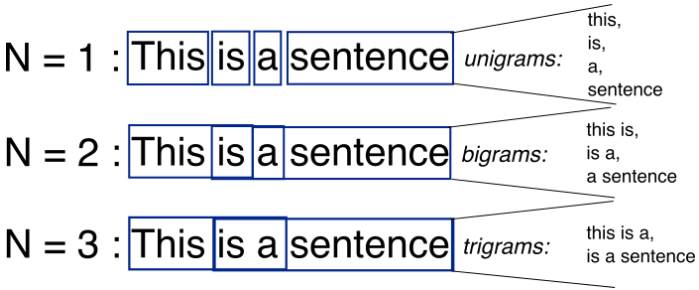
Q24. Which types of features for ML-based NLP algorithms leverage the concept of distributional semantics?

- Word-shape features.
- Context features, e.g. left or right token.
- Word Embeddings.
- One-hot-encoded word identity.

More Token Features

- Dictionary match
- Part-of-speech tags (e.g., is the word a proper noun)
- Context features
  - Distributional semantics: “a word is characterized by the company it keeps” (J.R. Firth, 1957)
    ➔ Surrounding tokens can be more important than token itself
  - word n grams (common: trigrams with current, left and right word)



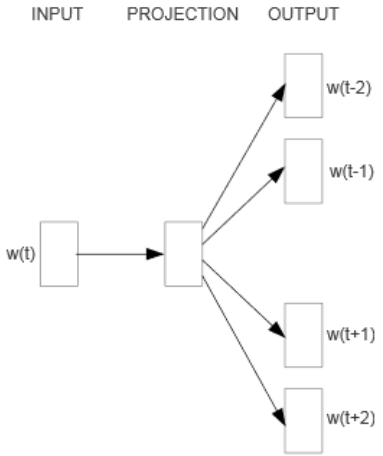


https://medium.com/analytics-vidhya/pos-tagging-using-conditional-random-fields-92077e5eaa31
https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram

Word Embeddings

- One-hot-vectors have no notion of semantic similarity
- Mikolov et. al (2013) popularized Word Embeddings
- Idea:
  - Train a feed-forward NN that predicts *surrounding words* (distributional semantics)
  - Use hidden layer as dense representation (projection) of words in a feature space that perform well at this task
- self-supervised → no labels required
- Embeddings are used as feature vectors for downstream tasks, like NER (= transfer learning)

$$v(\text{“cancer”}) \rightarrow \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} \quad v(\text{“carcinoma”}) \rightarrow \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 0 \\ 1 \\ \dots \\ 0 \end{pmatrix} \quad v(\text{“apple”}) \rightarrow \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



Mikolov et. al (2013): Efficient Estimation of Word Representations in Vector Space

Q24. Which types of features for ML-based NLP algorithms leverage the concept of distributional semantics?

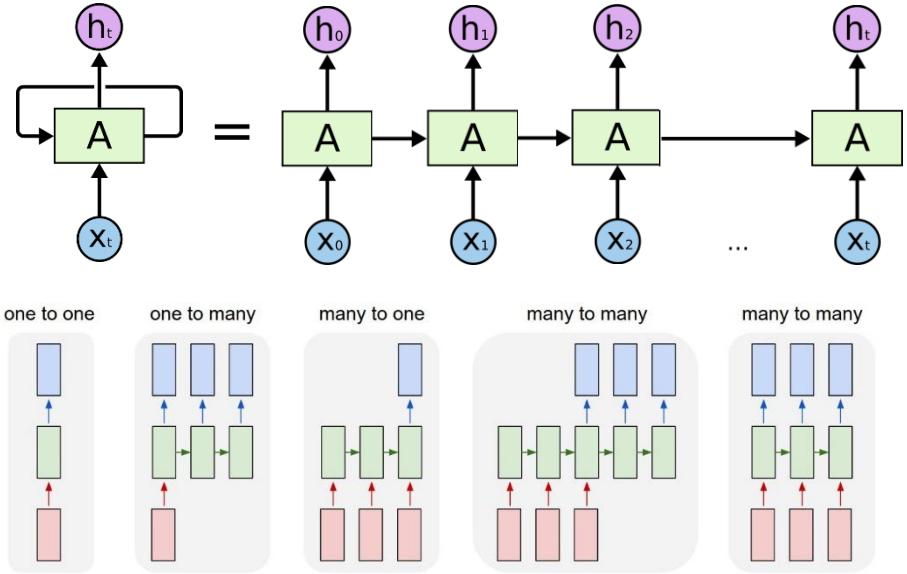
- Word-shape features. 
- Context features, e.g. left or right token. 
- Word Embeddings. 
- One-hot-encoded word identity. 

Q25. Why would a deep-learning-based sequence-to-sequence model be preferred to a simple token classification approach for named entity recognition?

- Training sequence-to-sequence models is computationally more efficient than independent token classification.
- Deep learning models for sequence-to-sequence prediction allow to model the problem in an end-to-end fashion without solving intermediate steps such as feature engineering.
- Sequence-to-sequence approaches make fewer assumptions about statistical independence of adjacent token labels.
- Recurrent neural networks are interpretable by design as opposed to classification algorithms, such as Logistic Regression.

Recurrent Neural Networks

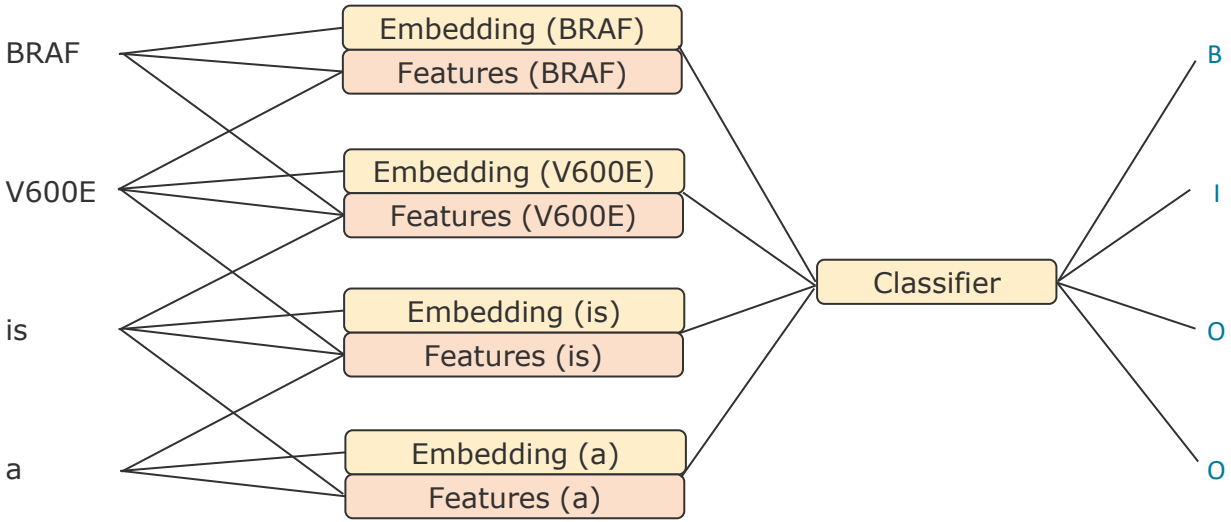
- Special type of neural network with recurrent connections → variable number of computational steps
- Various types of RNN cells exist (denoted by in t[A]cture)
- Enable end-to-end learning := directly map inputs (sentences) to outputs without explicitly solving intermediate steps (feature engineering, lingustic analysis)
- Computationally much more demanding than linear models



https://colah.github.io/posts/2015-08-Understanding-LSTMs/
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Structured Prediction for NLP

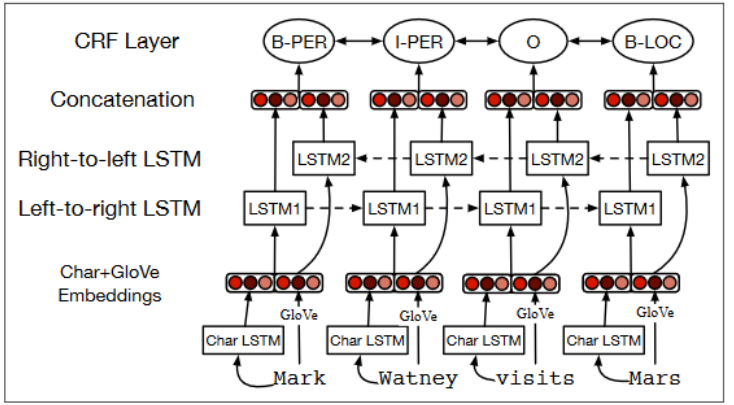
- Classifying single tokens assumes statistical independence of adjacent labels ➔ usually wrong
- Idea: predict whole sequence of labels **y** from sequence of inputs **x** (Structured Prediction)



- … in practice we need to impose some restrictions to keep the problem tractable

Deep Learning for Sequence-to-Sequence Problems

- State-of-the art approach for NER is (used to be?) bi-LSTM-CRF:
  - Word embeddings pretrained on large, unlabelled corpus (such as PubMed abstracts)
  - Bidirectional Long-Short-Term-Memory (LSTM) neural network (special form of recurrent neural network) for feature learning
  - CRF output layer to predict sequence of labels from LSTM representations
- Very recent neural architectures are based on Transformers instead of LSTMs (BERT, OpenAI GPT)
- Sequence-to-sequence architectures also applicable to other problems, such as translation or summarization



Martin, James H., and Daniel Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. 3rd ed. Draft https://web.stanford.edu/~jurafsky/slp3/*

Google's Neural Machine Translation System

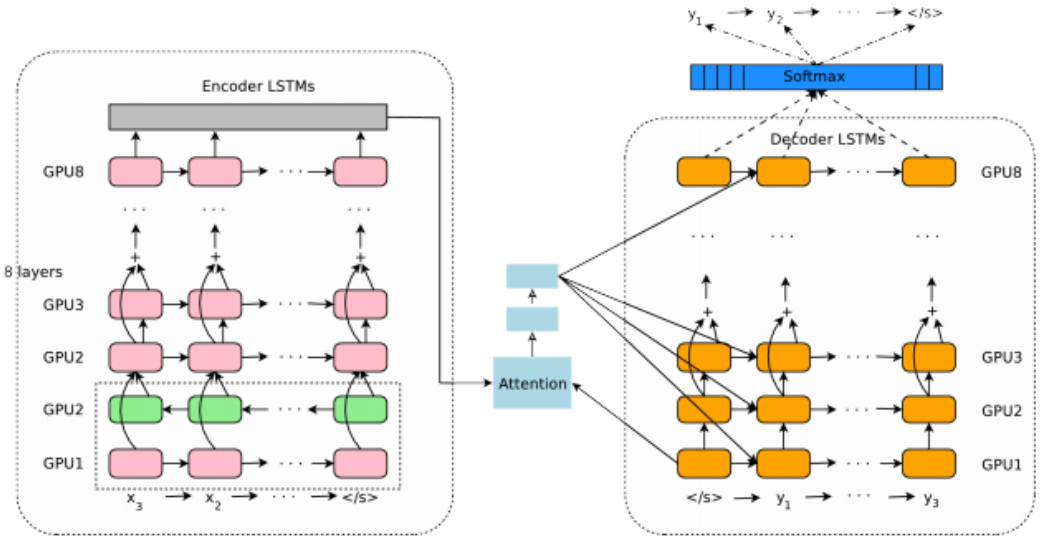


Figure 1: The model architecture of GNMT, Google's Neural Machine Translation system. On the left is the encoder network, on the right is the decoder network, in the middle is the attention module. The bottom encoder layer is bi-directional: the pink nodes gather information from left to right while the green

Q25. Why would a deep-learning-based sequence-to-sequence model be preferred to a simple token classification approach for named entity recognition?

Hasso Plattner Institut

- Training sequence-to-sequence models is computationally more efficient than independent token classification. 
- Deep learning models for sequence-to-sequence prediction allow to model the problem in an end-to-end fashion without solving intermediate steps such as feature engineering. 
- Sequence-to-sequence approaches make fewer assumptions about statistical independence of adjacent token labels. 
- Recurrent neural networks are interpretable by design as opposed to classification algorithms, such as Logistic Regression.