Lecture 2: Electronic Medical Records

CSCI6410/EPAH6410/CSCI4148

Finlay Maguire (finlay.maguire@dal.ca)

Learning Outcomes

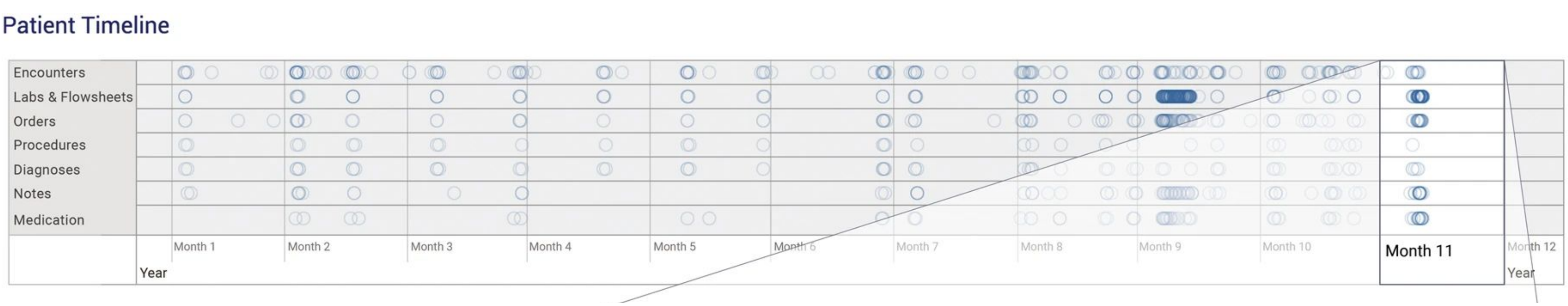
- Describe electronic medical/health record systems and the types of data they typically contain
- Distinguish structured, semi-structured, unstructured text data
- Describe approaches to searching text
- Outline key steps in preparing text for analysis
- Explain the general concept of learnt word embeddings
- Explain how embeddings can be tuned/customised
- Identify differences between named entity recognition, parts of speech tagging, and dependency parsing

- ***Not covered: fuzzy search and text indexing***

What is an Electronic {Medical,Health} Record?

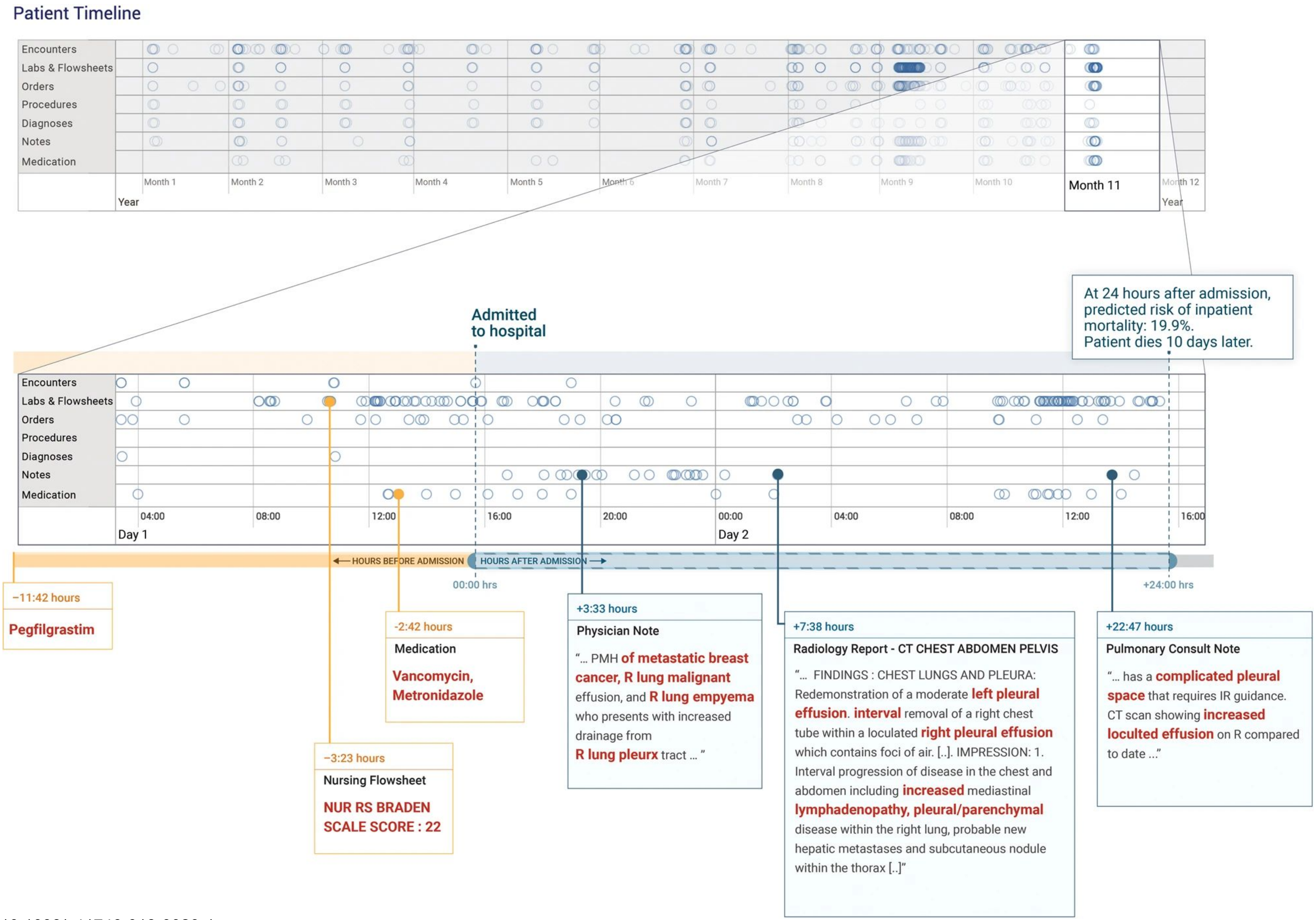
EMR are digital patient charts

- Repository of patient information over time
- Prone to fragmentation between primary / hospital care
- Ideally contains all of a given patient's details on:
    - Every encounter with health professionals (e.g., admitted to hospital)
    - Details and results of diagnostic testing and vitals (e.g., blood test, urine cultures etc.)
    - Diagnostic/therapeutic orders (e.g., Nil per os/NPO,
    - Procedures performed (e.g., appendectomy, PET-scan)
    - Medical note (e.g., primary physician, consult information)
    - Medication (e.g., antibiotics)



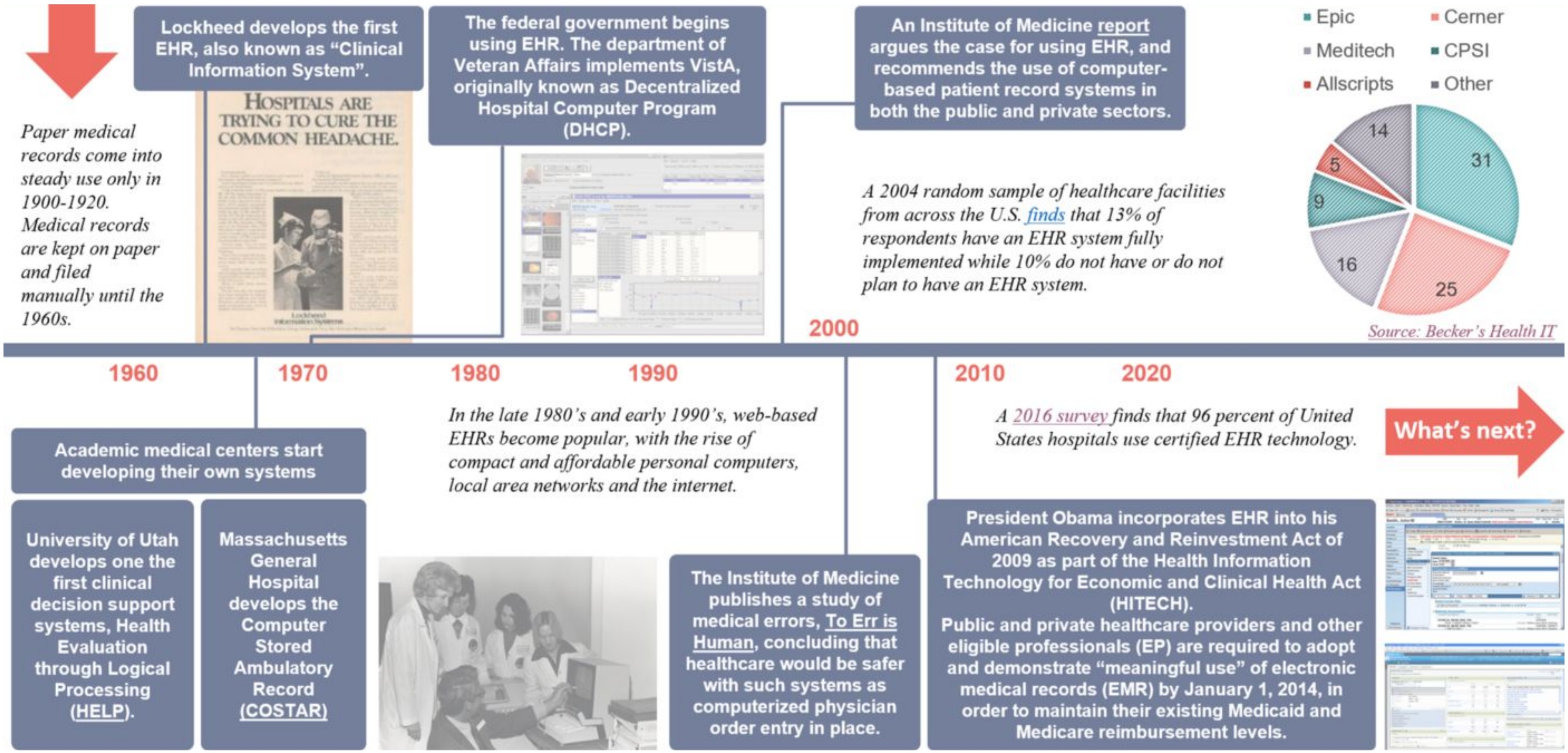
10.1038/s41746-018-0029-1

EMR data varies in structure



10.1038/s41746-018-0029-1

EMRs have a surprisingly long history



https://mayaberlerner.medium.com/a-brief-history-of-ehrs-c51a2125a247

EMRs are common and increasing in use in Canada

- Use of EMRs is common and increasing in Canada (2017-2024: 82% to 95%)
- Primary care is main users (97%) compared to specialists (93%)
- Atlantic regions have lowest use rate (86%)
- Main features used:
    - Ordering diagonstic tests / accessing results (72%)
    - Prescription system with automatic warning of adverse drug interactions (60%)
    - Communication of discharge/consultation notes (40%)
- Not used:
    - Clinical decision support (<27%)
    - Appointment scheduling (<25%)

EMR System	National	Nova Scotia
Accuro - QHR	15.1%	32.0%
Med Access - Telus	13.1%	38.5%
PS Suite - Telus	12.8%	0%
Epic	8.0%	0%
Secure EHR - Citrix	5.6%	0%
Meditech	5.6%	20.3%
Oracle Cerner	1.5%	0%
Other	11.7%	9.2%

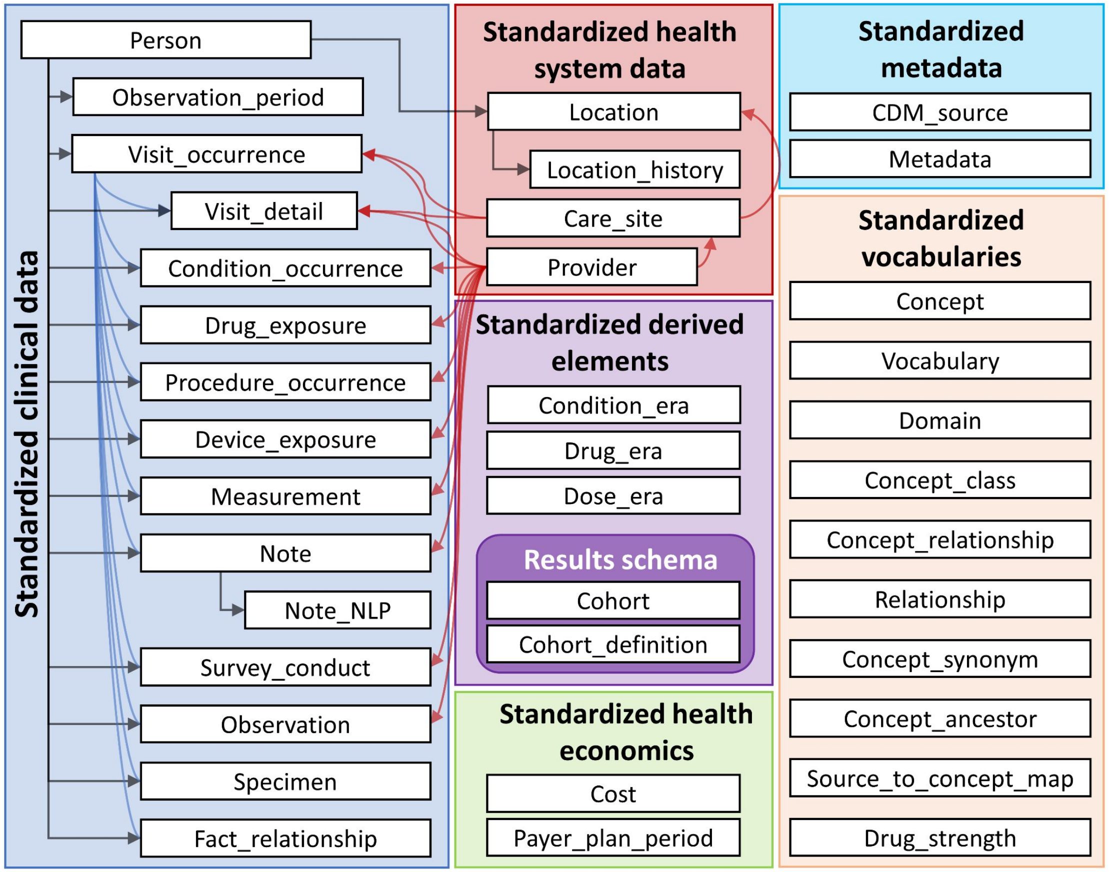
2021 & 2024 National Survey of Canadian Physicians

One Person ~~Patient~~ One ~~Record~~ ~~Experience~~ Record

- Mix of Meditech, SHARE, individual silos, 80+ systems (HCWs use 5 on average per patient)
- One Person One Record
  - 2014 December: Strategy Approval
  - 2015 July:
    - NSHA CIO appointed to lead project
    - RFP solutions hired to monitor procurement process to lead OPOR
  - 2016 March: Meeting moratorium
  - 2016 May/December: Gevity/Allscripts NSHA meetings
  - 2017 January: Request for Supplier Qualifications
  - 2017 February: Submissions from 4 big firms (**Epic**, Allscripts, Cerner, **Meditech**) and 2 small ones (Evident, Harris Healthcare Group)
  - 2017 June: Allscripts and Cerner named as only finalists based on 50 page RFSQ
  - 2017 August: Evident complaint
  - ....
  - ???
  - …
  - 2022: rename to One Patient One Experience
  - 2023: rename to One Person One Record & 10-year $365M contract with **Oracle Health** (formerly Cerner)
  - 2025 February: Planned partial roll-out
  - 2025 May: Delay to planned roll-out & new physician lead hiring…
  - 2025 December: New rollout planned date

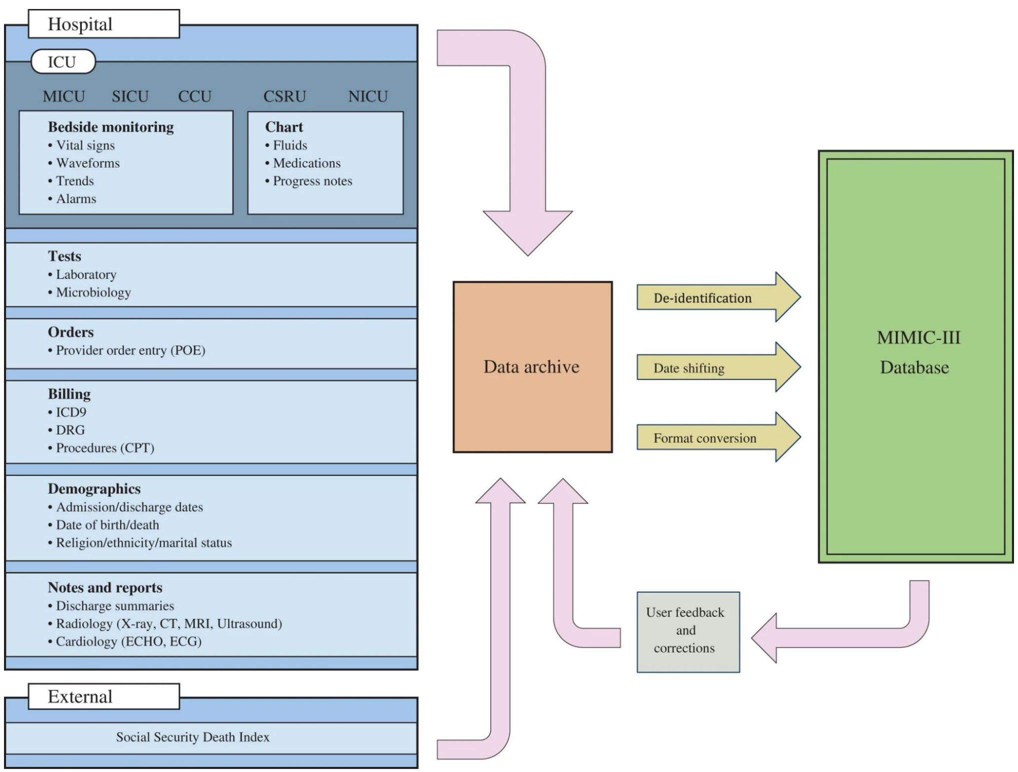
Reality: fragmented EMRs requiring difficult linkage

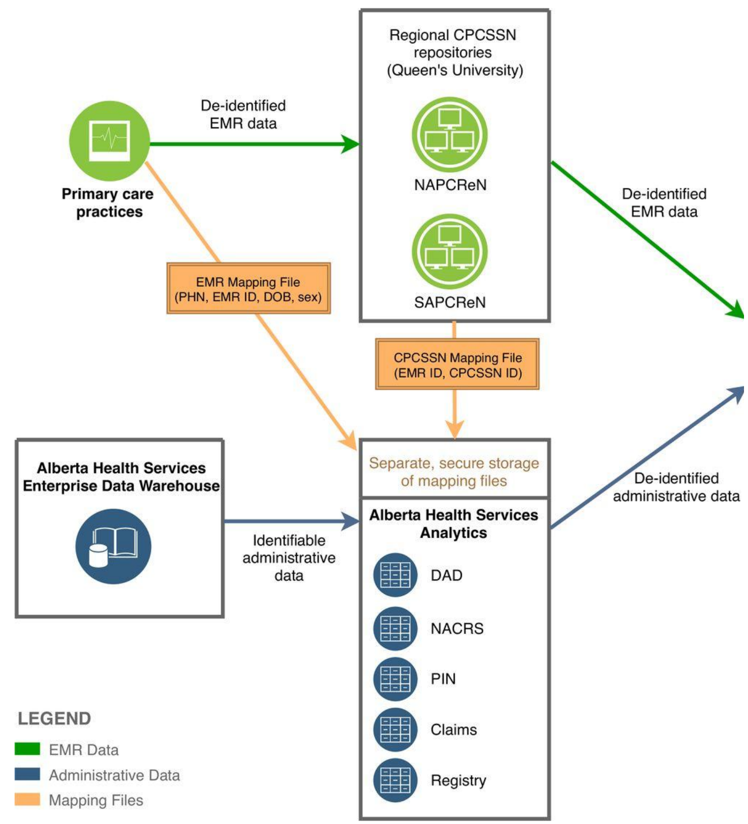
- Interoperability is a competitive disadvantage
- Standarised format and vocabulary for EMR data: Observational Medical Outcomes Partnership (OMOP) common data model
- Observational Health Data Sciences and Informatics (OHDSI) tooling
- Fast healthcare interoperability resources (FHIR)



https://ohdsi.github.io/TheBookOfOhdsi/images/CommonDataModel/cdmDiagram.png

EMR datasets: MIMIC-.* / Canadian Primary Care Sentinel Surveillance Network / STARR

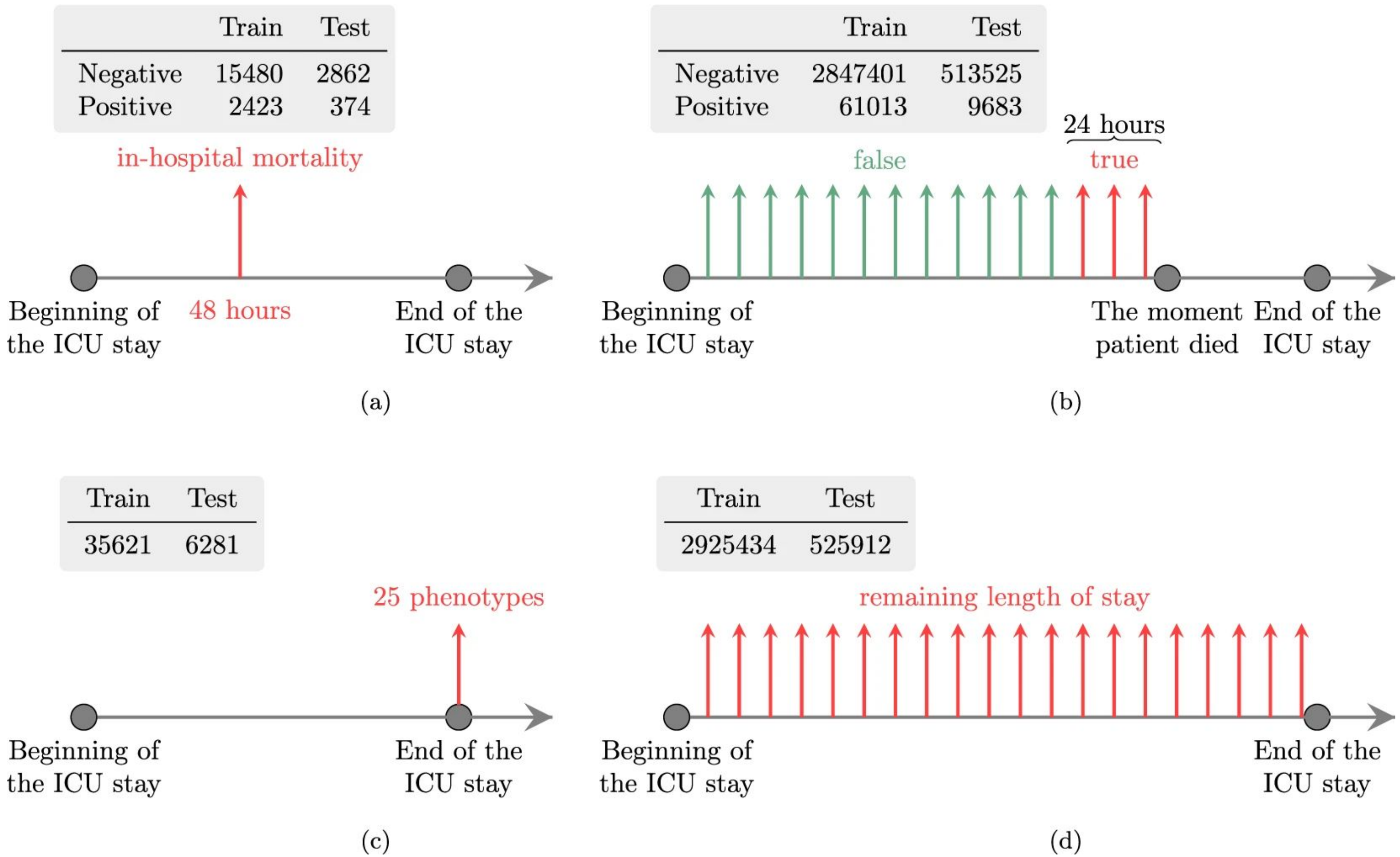




10.1038/sdata.2016.35

https://informatics.bmj.com/content/27/3/e100161

EMR allow us to ask complex questions



What kind of data is in an EMR?

Many types of data in EMRs

- Discrete Physiological Parameters e.g., blood test metric measures
- Diagnostic Imaging Data e.g., MRI image data
- Physiological Sensor Data e.g., EKG/EEG signal data
- Ordinal scale assessments e.g., frailty index

Text:

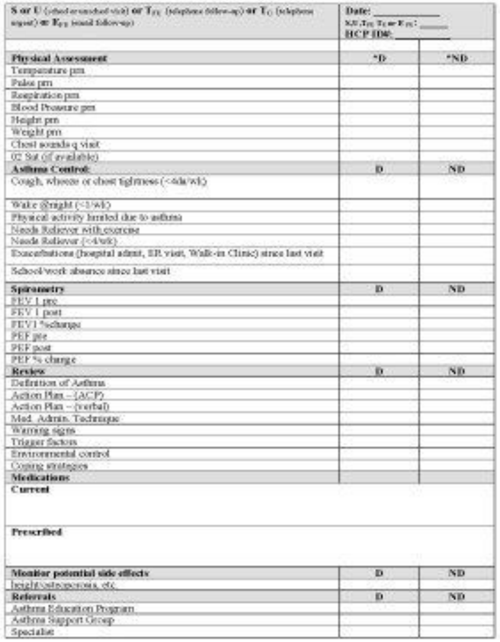
- <u>Structured text</u> e.g., CPT/ICD-10 codes (*V89.2XXA, S06.0)*
- <u>Semi-structured Text</u> e.g., *{"Symptoms": "Head pain, dizziness, emesis", "Cause": "Car crash", "Diagnosis": "Likely concussion"....}*
- <u>Unstructured Text</u>: *"Patient was involved in a car crash and presented to the ER with pain, vomiting, and mild dizziness*. *Most likely they are concussed but should follow up with a head CT to confirm no other brain injuries"*

CAT SCANS	
ABDOMEN	
Abdomen w/o contrast	74150
Abdomen w/ contrast	74160
Abdomen w/o & w/ contrast	74170
CHEST/THORAX	
Chest/Thorax w/o contrast	71250
Chest/Thorax w/ contrast	74150
Chest/Thorax w/o & w/ contrast	71270

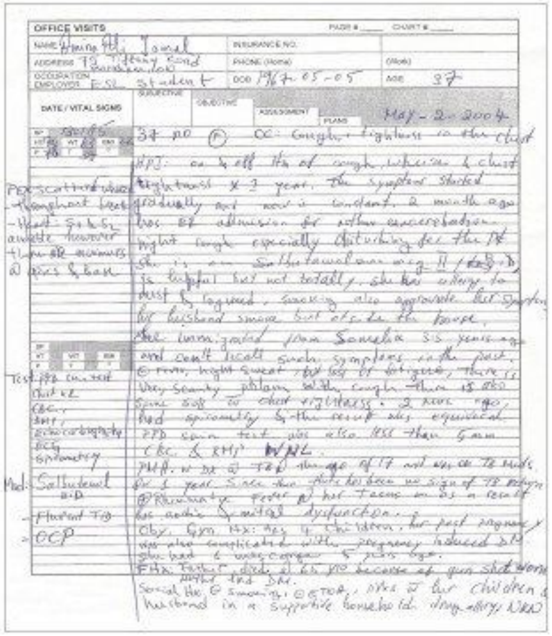
	ICD-10	DESCRIPTION
ormations and Chromosomal Abnormalities (Including Down's Syndrome)		
EC	Q64.79	Other congenital malformations of bladder and urethra
	Q90.9	Down syndrome, unspecified
IOS	Q05.4	Unspecified spina bifida with hydrocephalus
	Q05.8	Sacral spina bifida without hydrocephalus
nitourinary System Diseases (Including Incontinence and UTI)		
IOS	N17.9	Acute kidney failure, unspecified

Medicine loves unstructured text

- Unstructured text is and will likely forever remain the primary form of communication in medical clinical settings.
- Highly flexible, efficient, and expressive across a range of communication contexts for medicine.
- Mainstay of charts, notes, consults, discharge summaries, procedure/operative logs.



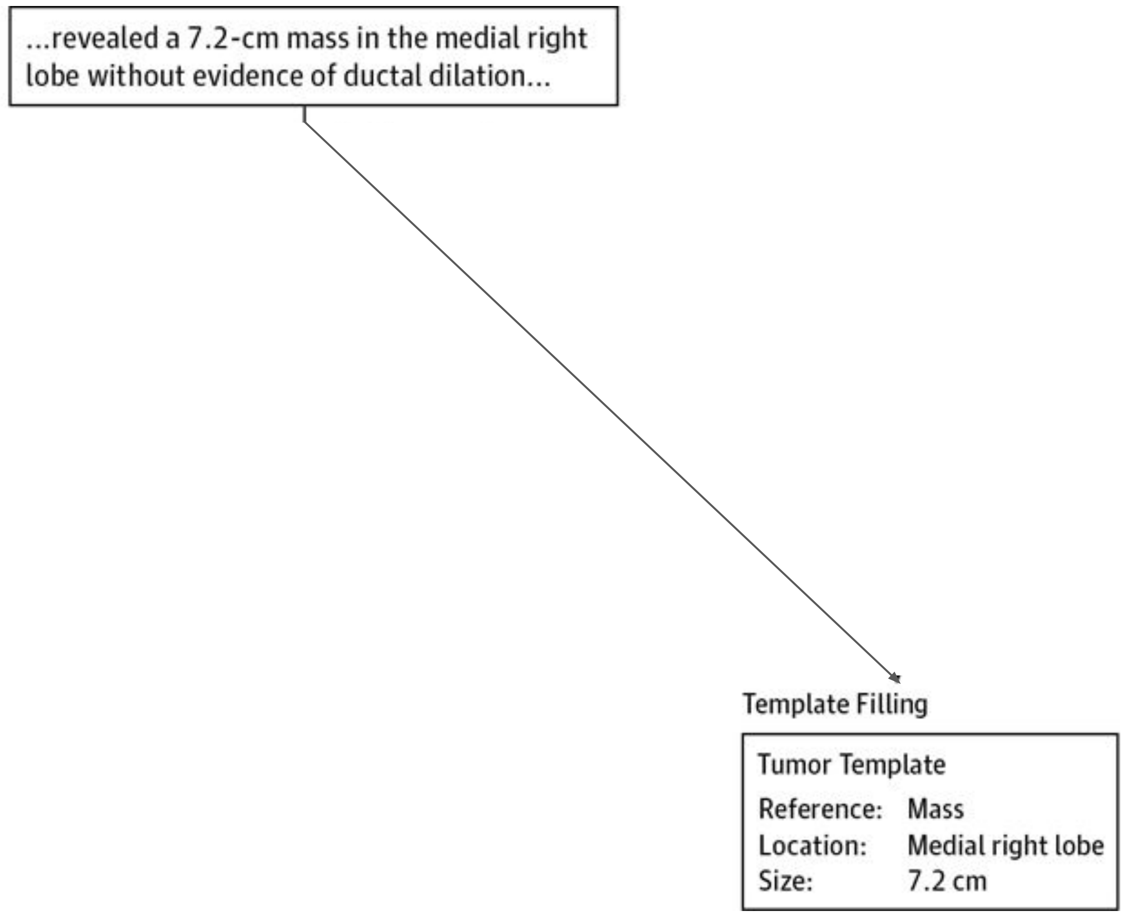
S or U (scheduled or unscheduled visit) or T_{FU} (telephone follow-up) or T_U (telephone urgent) or E_{FU} (email follow-up)	Date: ___ S/U, T_{FU} T_U or E_{FU}: ___ HCP ID#: ___	
Physical Assessment	*D	*ND
Temperature prn		
Pulse prn		
Respiration prn		
Blood Pressure prn		
Height prn		
Weight prn		
Chest sounds q visit		
O2 Sat (if available)		
Asthma Control	D	ND
Cough, wheeze or chest tightness (<4d/wk)		
Wake @night (<1/wk)		
Physical activity limited due to asthma		
Needs Reliever with exercise		
Needs Reliever (<4/wk)		
Exacerbations (hospital admit, ER visit, Walk-in Clinic) since last visit		
School/work absence since last visit		
Spirometry	D	ND
FEV 1 pre		
FEV 1 post		
FEV1 %change		
PEF pre		
PEF post		
PEF % change		
Review	D	ND
Definition of Asthma		
Action Plan – (ACP)		
Action Plan – (verbal)		
Med. Admin. Technique		
Warning signs		
Trigger factors		
Environmental control		
Coping strategies		
Medications		
Current		
Prescribed		
Monitor potential side effects	D	ND
height/osteoporosis, etc.		
Referrals	D	ND
Asthma Education Program		
Asthma Support Group		
Specialist		



Unstructured text is challenging

- English language especially has many synonymous and highly flexible grammatical structure.
- Medical english has many synonyms and similar seeming non-synonyms:
    - Bilateral salpingectomy
    - Salpingectomy
    - Fallopian Transection
    - Fallopian Tubectomy
    - Fallopian Tubal Ligation
    - Tubal ligation
    - Tubal sterilisation
    - Tubal
    - CPT58600
- Now add typos and transcription errors!
- Difficult to search
- Difficult to summarize
- Difficult to analyze

More complex operations require additional steps



10.1001/jamaoncol.2016.0213

So how would we do something like this?

Natural Language Processing!

- NLP is any computer-based method that handles/augments/transforms natural language so that it can be represented for computation.
- Approximate synonyms: text mining, text processing, computational linguistics

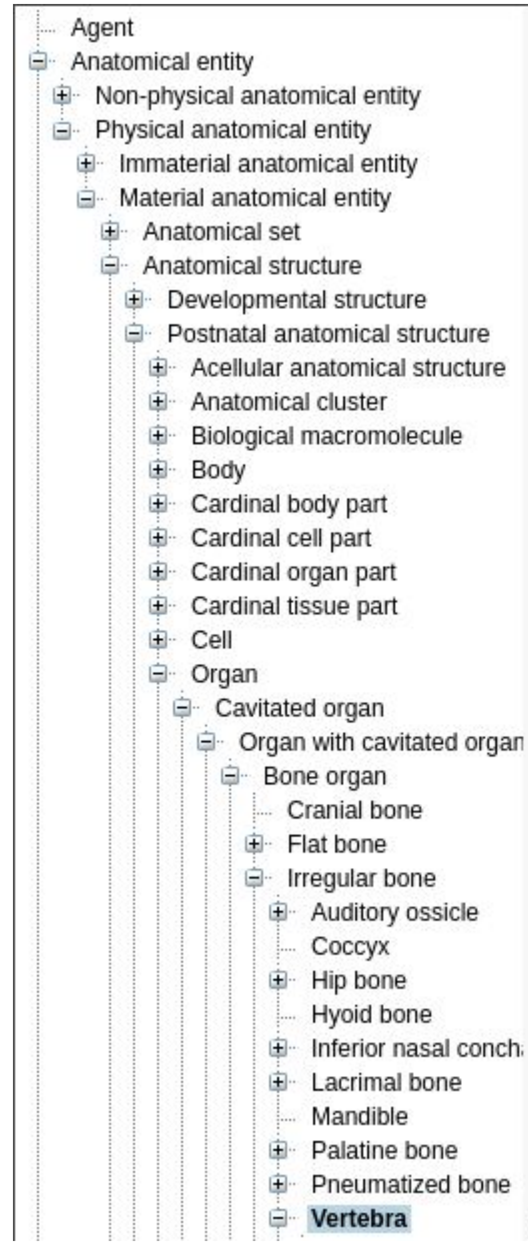
- Example problem:

  1. “Find every medical note in the EMR related to the spine”

  2. Identify key search terms e.g., “back”, “spine”, “vertebra”, “lumbar”, “neck”, “cervical”, “thoracic”, “sacrum”, “coccyx” (expertise, ontology/vocabulary)

  3. Search for EMR for these terms

MESH

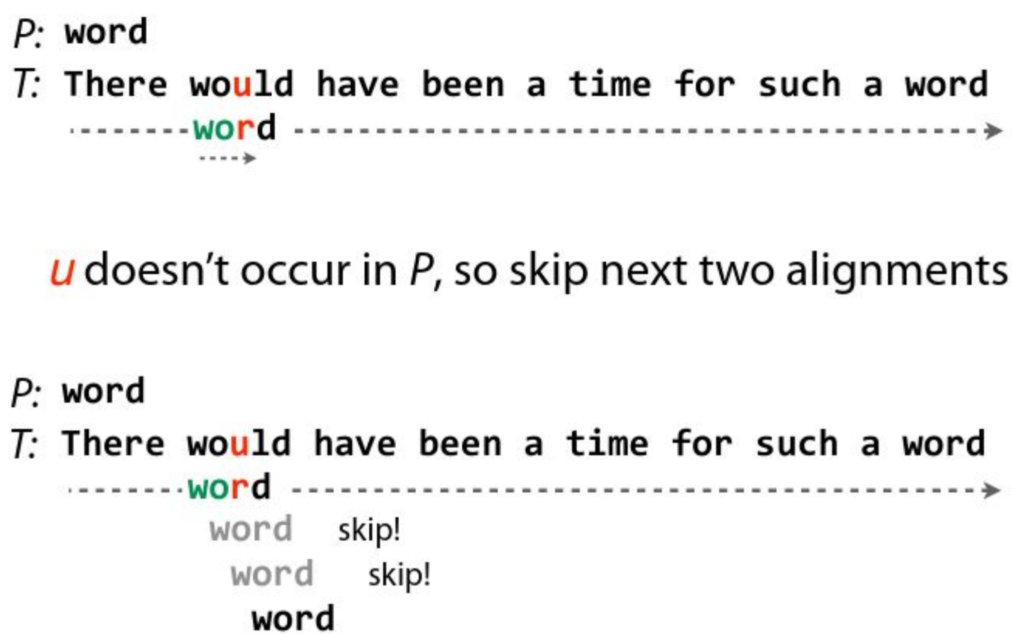


Let’s start simple: searching text

Searching for exact matches: Ctrl-F

Many exact match algorithms with varied properties (typically ctrl-F will mix and match them in a context-dependent way).

- Scan over all text and look for things that exactly match your query
- Make things more efficient: Boyer-Moore/Knuth-Morris-Pratt/Rabin-Karp etc.



https://www.cs.jhu.edu/~langmea/resources/lecture_notes/04_boyer_moore_v2.pdf

More flexible searches for keywords: Regular Expressions

- Need to find “spine” and “spinal” = *spin*(*a*|*e*)*l*?
- Can also be used to capture words/before after: **\w+\s***spin*(*a*|*e*)*l*?**\s\w+**
- Builds on lots of well-developed CS theory

- *You have a problem, you use regex, you now have 2 problems*

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
$	Ends with	"world$"
*	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
{}	Exactly the specified number of occurrences	"al{2}"
|	Either or	"falls|stays"
()	Capture and group	

Regular expressions can get very complicated!

RCF5322 Email validation regex:

```
(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08
\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?
:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:(2(5[
0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0
-9]|[1-9]?[0-9])|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\
x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])
```

Can we make the text easier to search instead?

Most NLP methods start with text normalisation

1. Tokenisation
2. Normalising word formats
3. Segmenting sentences

Splitting text into words: Segmentation/Tokenizing

- Breaking text into individual units (letters/morpheme/words/sentences/paragraphs) can make it much easier to handle.
- Process is known as tokenisation (a subset of segmentation)
- Units you break into are known as tokens:

“*Indication of significant spinal contusions*.”

-> “*Indication” “of” “significant” “spinal” “contusions*.”

- Easy approach:
    - split on spaces
    - Has to be fast (finite state automata)
- Challenges: punctuation can matter (e.g., 01/02/22), not all languages use spaces, may want to treat multiword expressions (MWE) as tokens e.g., “New York”, “bilateral salpingo-oophorectomy”, “ice box”/”ice-box”/”icebox”

Simplifying language: word normalisation

- “Ph.D.”, “PhD”, “phd” probably shouldn’t be counted differently
- Case folding: collapse everything to lowercase (although case can often be informative: “US” vs “us”)
- Lemmatization: identifying words with common root (lemma) e.g., “operation” and “operations” -> “operation”; “am”, “are”, “is” -> “be”
  - “Surgeon is performing surgical procedures” -> “Surgeon be perform surgical procedure”
- Requires morphological parsing - splitting **stems** (central morpheme) from **affixes** (modifying/adidtional meaning)

- Lemmatization is difficult : alternative = stemming
  - Remove final affixes e.g., remove “-ing, -s, -ational, -sses”
  - “This was not the correct operation” -> “Thi wa not the correct operat”

Splitting sentences: sentence segmentation

- Sentences are delineated on punctuation: “.”, “?”, “!”

- Often we want to segment phrases/clauses, more challenging:
    - “Patient presented to ER with pain/confusion, most likely as a sequelae of a head injury” ->
        - [“Patient”, “presented”, “to” “ER”, “with”, “pain”, “confusion”]
        - [“most”, “likely”, “as”, “a”, “sequelae”, “of”, “a”, “head”, “injury”]

Hash-based text search

- Can find exact matches very efficiently
- Tokenize/lemmatise/normalise words in each note, then hash:
  - Note 1: [“spine”, “car”, “head”] -> [a11, a92, a53]
  - Note 2: [“car”, “tree”, “CT”] -> [a92, a57, a99]
- Hash query words “back”, “spine”, “vertebra”, “lumbar”, “neck”, “cervical”, “thoracic”, “sacrum”, “coccyx”:
  - a55, a11, …
- See if query hashes are present in note hash sets
  - a55 in Note1 = No, a55 in Note2 = No
  - a11 in Note1 = Yes, a11 in Note 2 = No

Can use these to create manual rules

if **finding** is in (“pneumothorax”; “hydropneumothorax”)

and **certainty-modifier** is not in

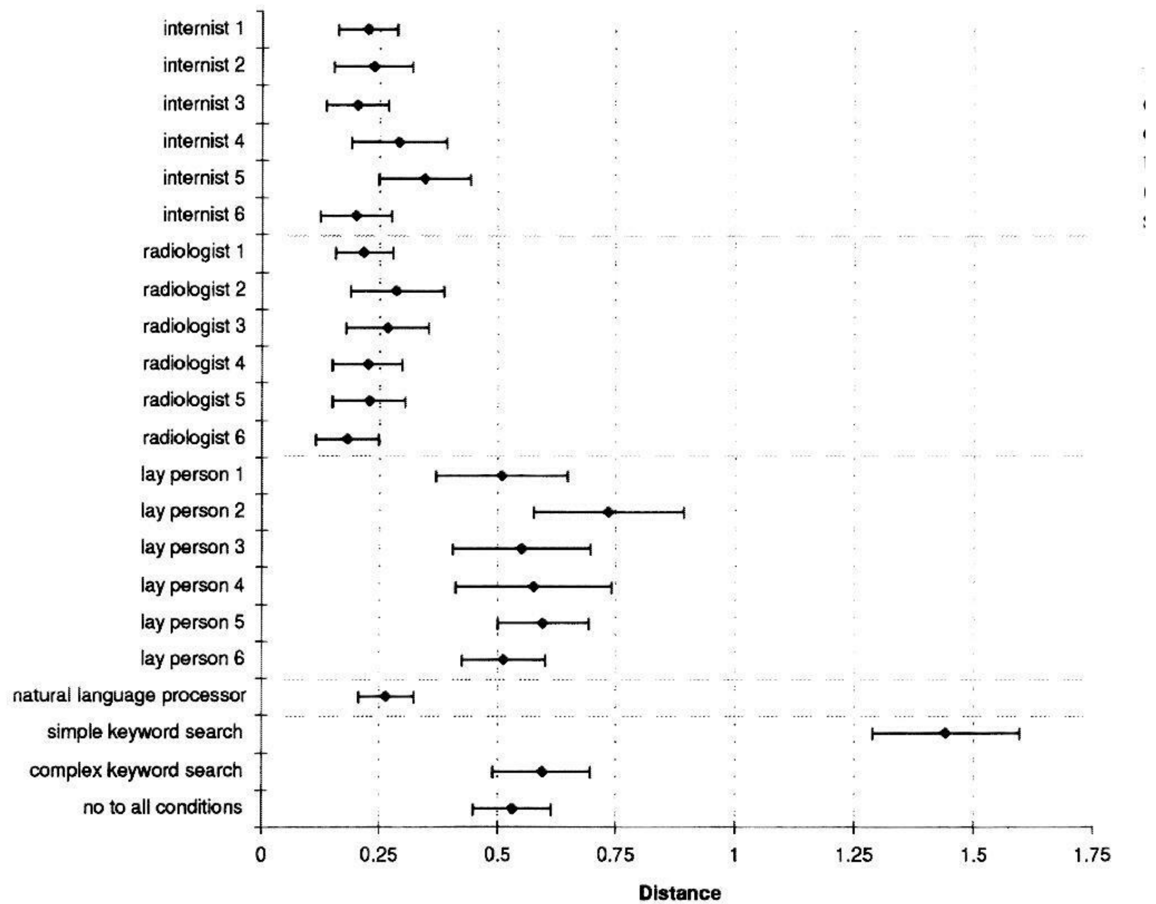
(“no”; “rule out”; “cannot evaluate”)

and **status-modifier** is not in (“resolved”)

then

conclude true;

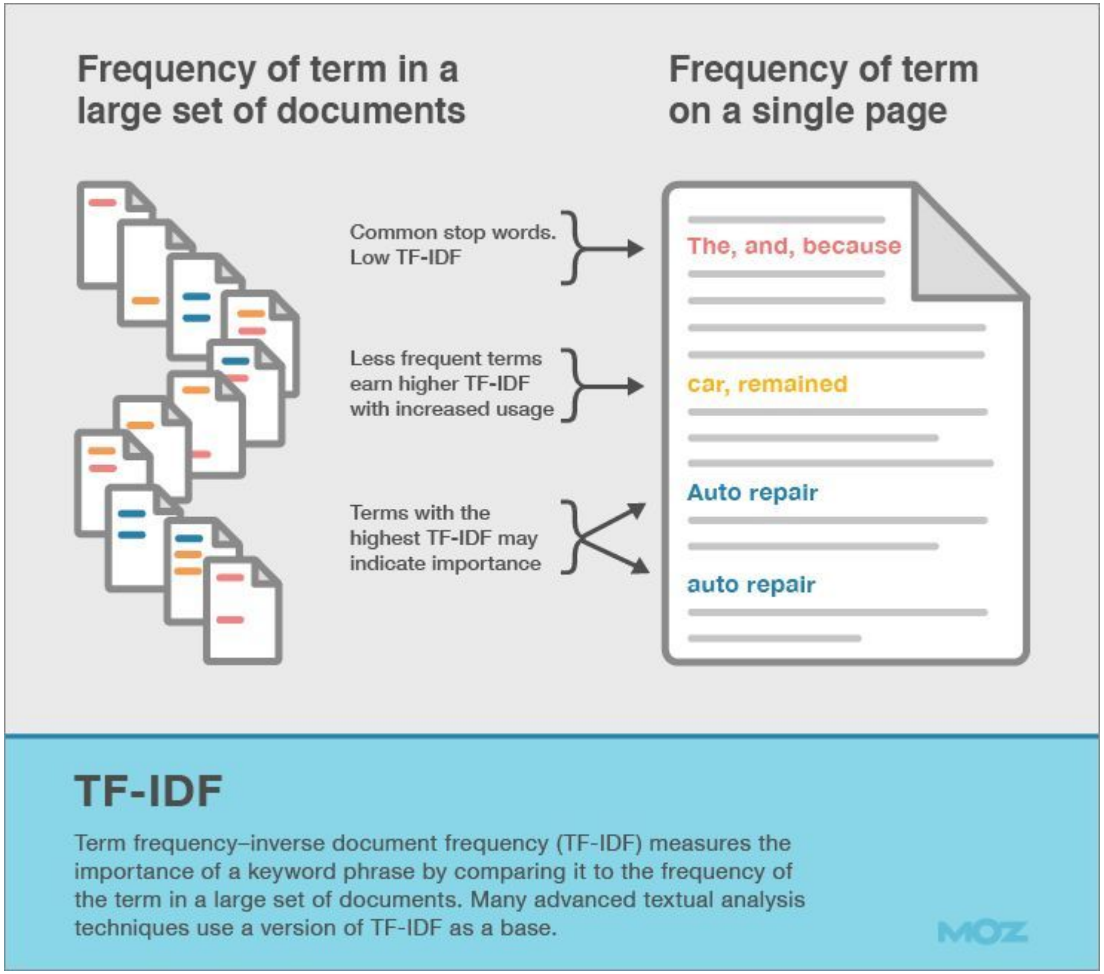
endif;



https://doi.org/10.7326/0003-4819-122-9-199505010-00007

What if we don’t know the query terms in advance?

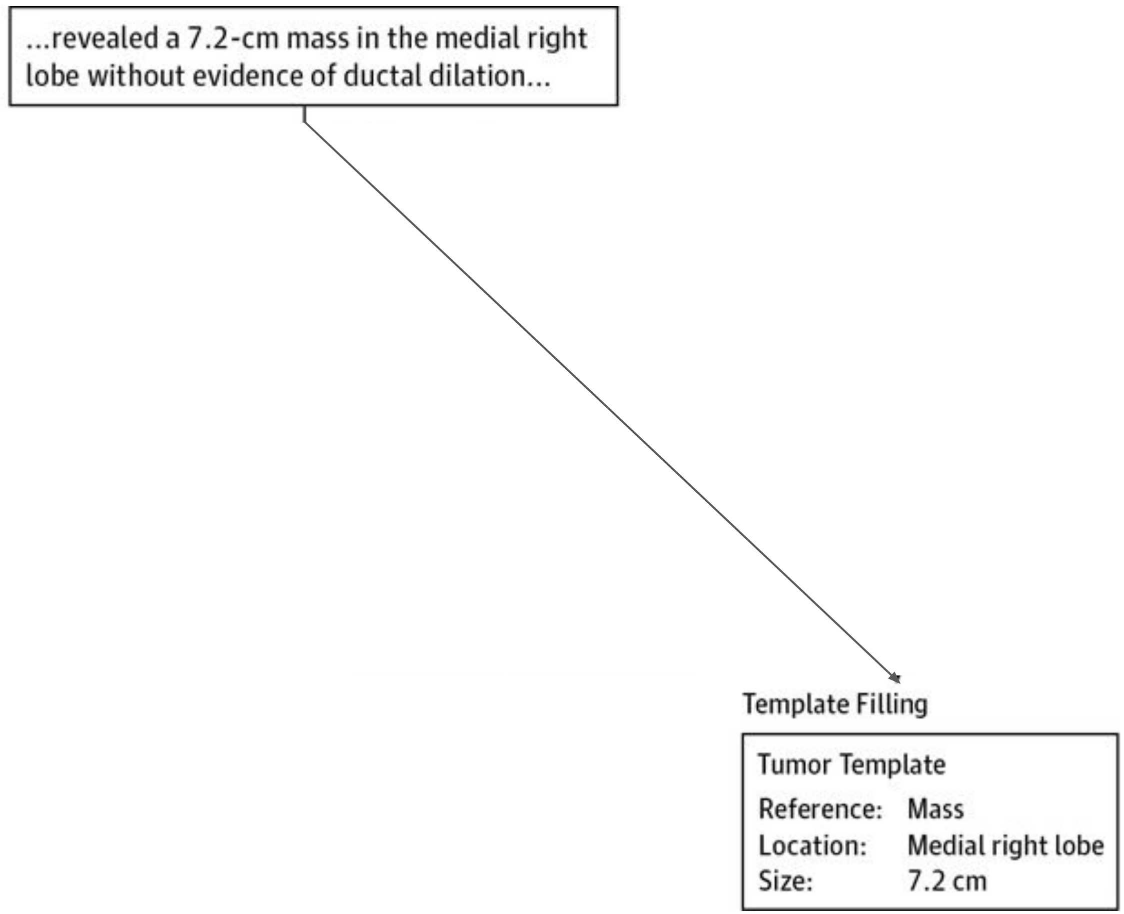
Identify frequently used terms



- Find highest TF-IDF terms
- Filter them manually for new search terms
- Apply prior search approaches (or any of the fuzzy matching approaches)
- Among other unsupervised approaches (e.g., following material)
- Matrix encoding using tf-IDF and/or co-occurrence

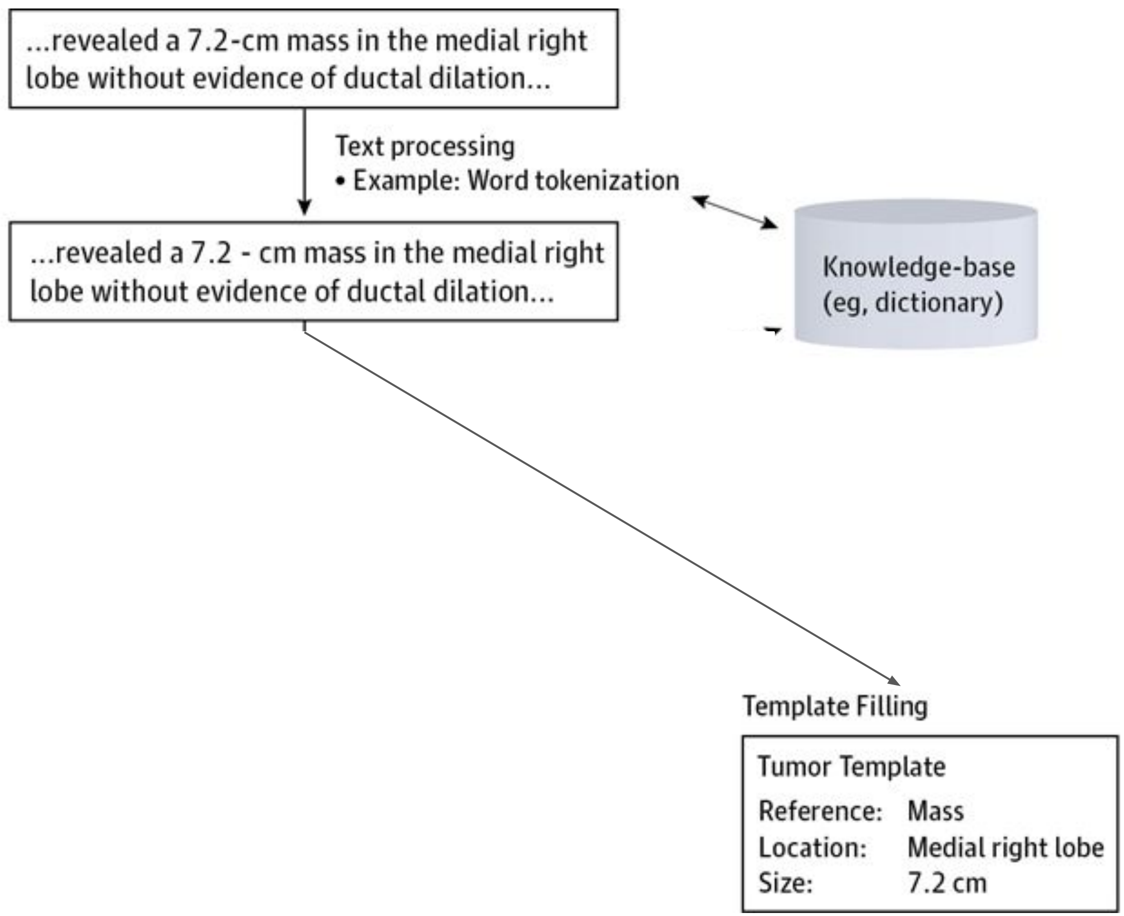
Can we automate more complex procedures?

More complex operations require additional steps



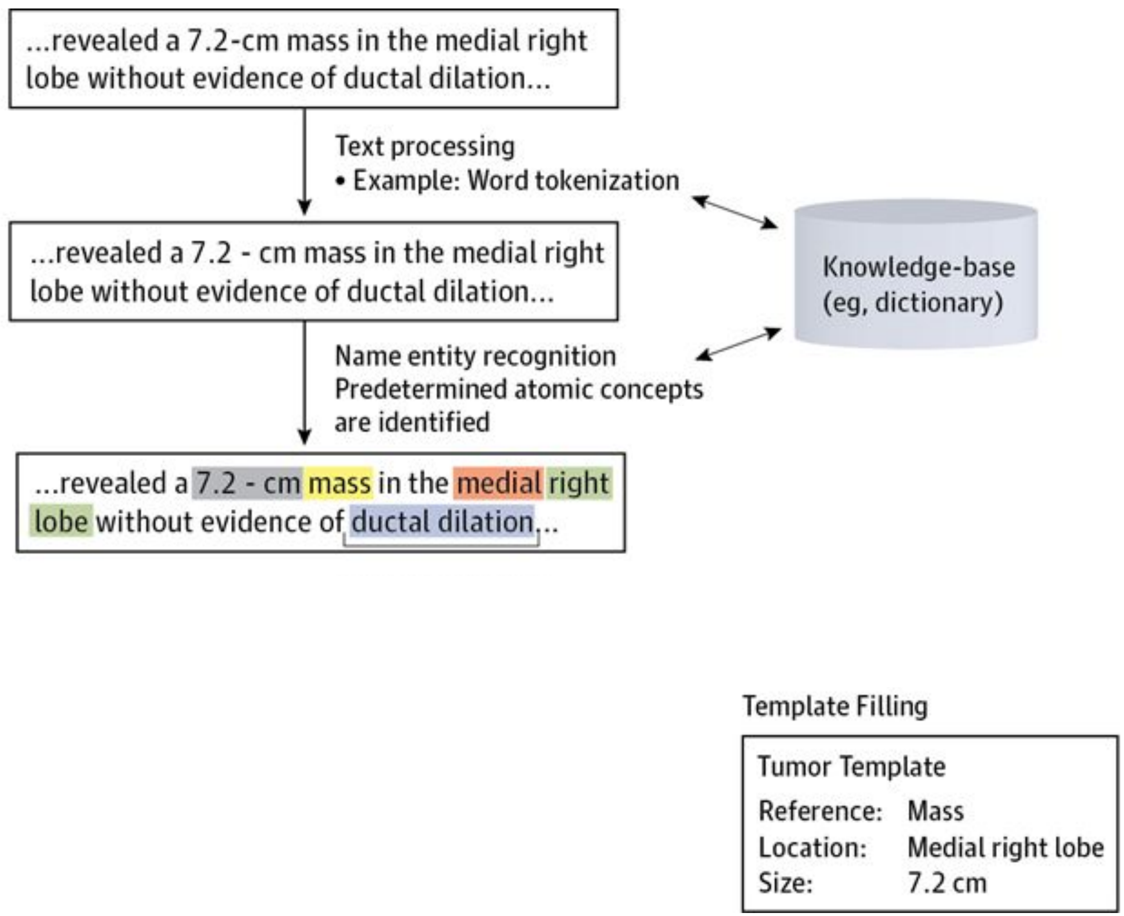
10.1001/jamaoncol.2016.0213

More complex operations require additional steps



10.1001/jamaoncol.2016.0213

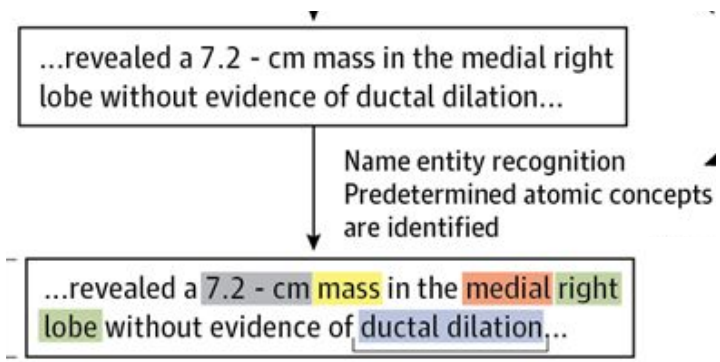
More complex operations require additional steps



10.1001/jamaoncol.2016.0213

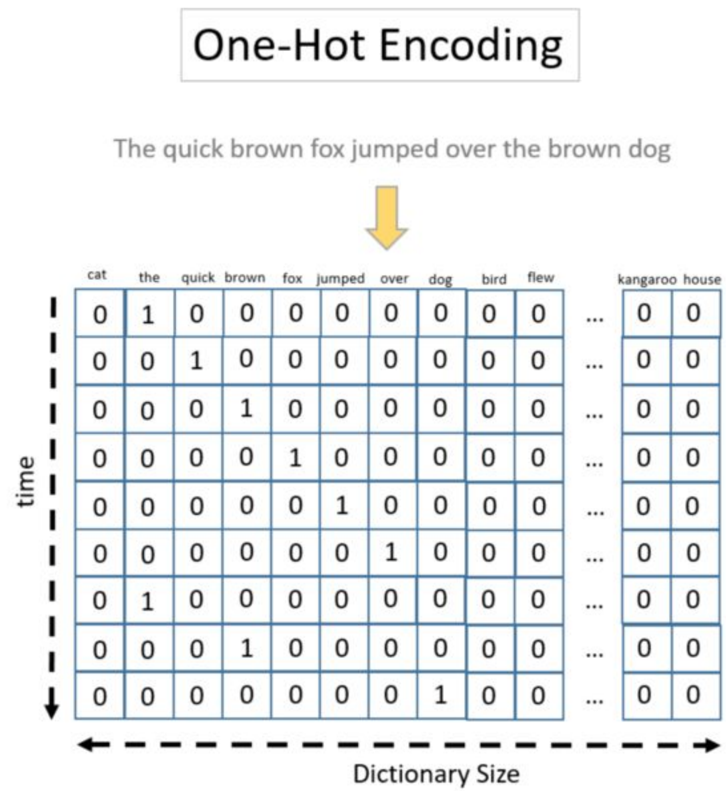
Identifying tokens referring to things: Named Entity Recognition

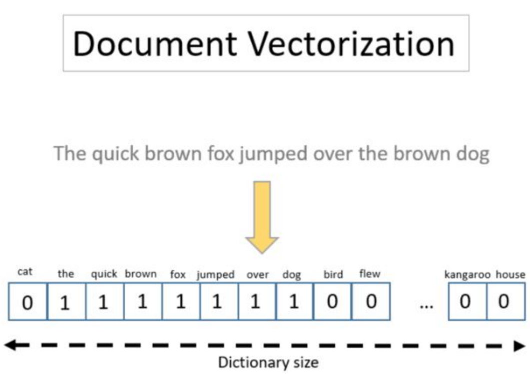
- Identify specific categories of entities e.g., places, times, anatomy
- Lots of pre-trained approaches/vocabularies
- Text classification problem => requires some way to encode text to a numerical vector



Encoding text as a vectors

- One-Hot encoding
- TF-IDF frequency based encodings
- Very large vectors with even moderate vocabulary size
- Very sparse vectors (lots of 0s)





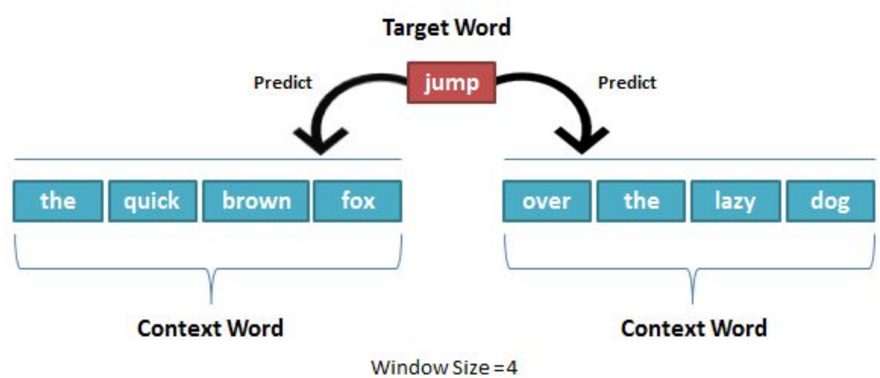
Sum over columns for each note to get a vector representation of the document instead (TF-IDF is a normalisation of this representation)

Reducing the dimensionality of these vectors

- Standard dimensionality reduction methods struggle
- Text has semantic (meaning) AND syntactic (grammar) components
- We want to find a lower dimensional embedding that captures these aspects

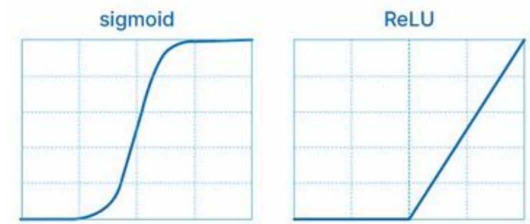
- “You shall know a word by the company it keeps” (Firth, J. R. 1957:11)
- “The meaning of a word is its use in the language” (Wittgenstein)
- Can we use the **CONTEXT** of a given word to find a meaningful vector representation?

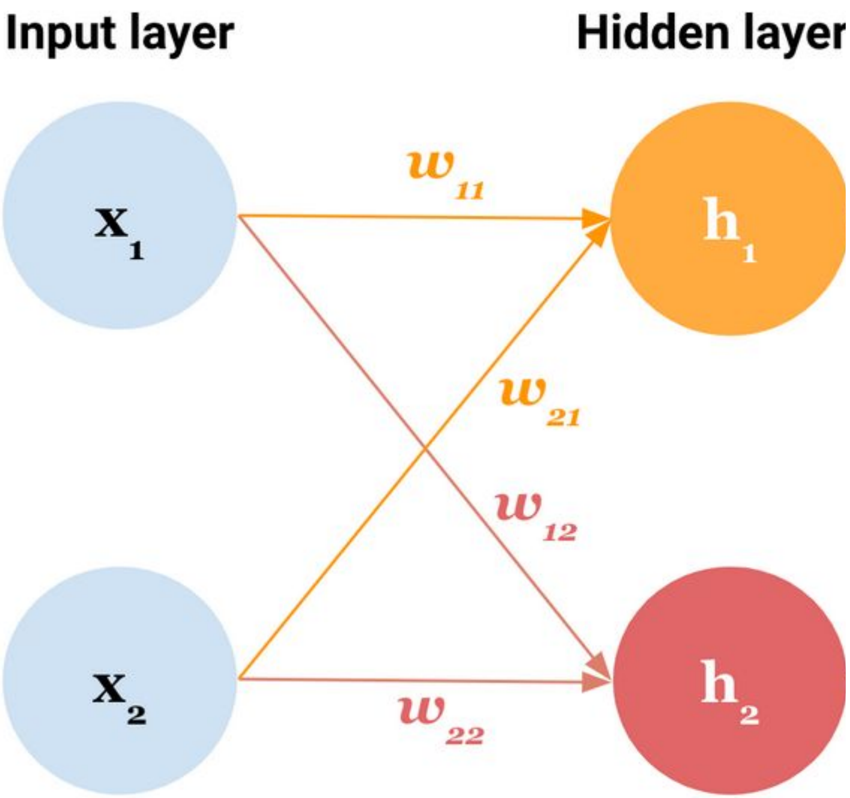
- Word2vec:
    - **Skip-Gram**:
        - Predict context from word
        - Better for rare words
    - **Continuous Bag of Words**:
        - Predict word from context
        - Better for common words



Neural Networks -

- Need: efficient models that capture high-dimensional non-linear relationships
- Solution: stack many simple models with a non-linearity (e.g., logistic / ReLU)
- **Neural Networks:**



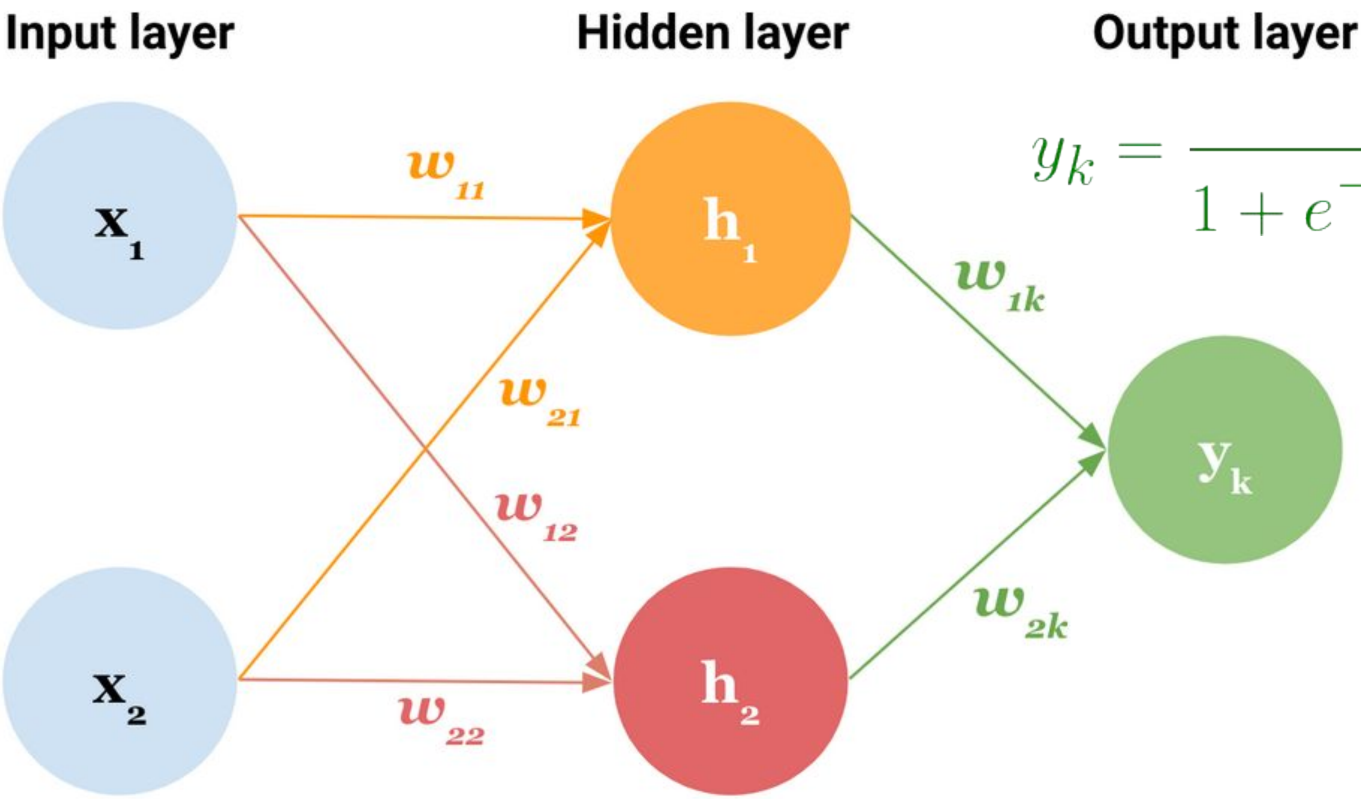


$$h_1 = \frac{1}{1 + e^{-w_{1,1}x_1 + w_{2,1}x_2}}$$

$$h_2 = \frac{1}{1 + e^{-w_{1,2}x_1 + w_{2,2}x_2}}$$

Neural Networks -

- Need: efficient models that capture high-dimensional non-linear relationships
- Solution: stack many simple models with a non-linearity (e.g., logistic / ReLU)
- **Neural Networks:**

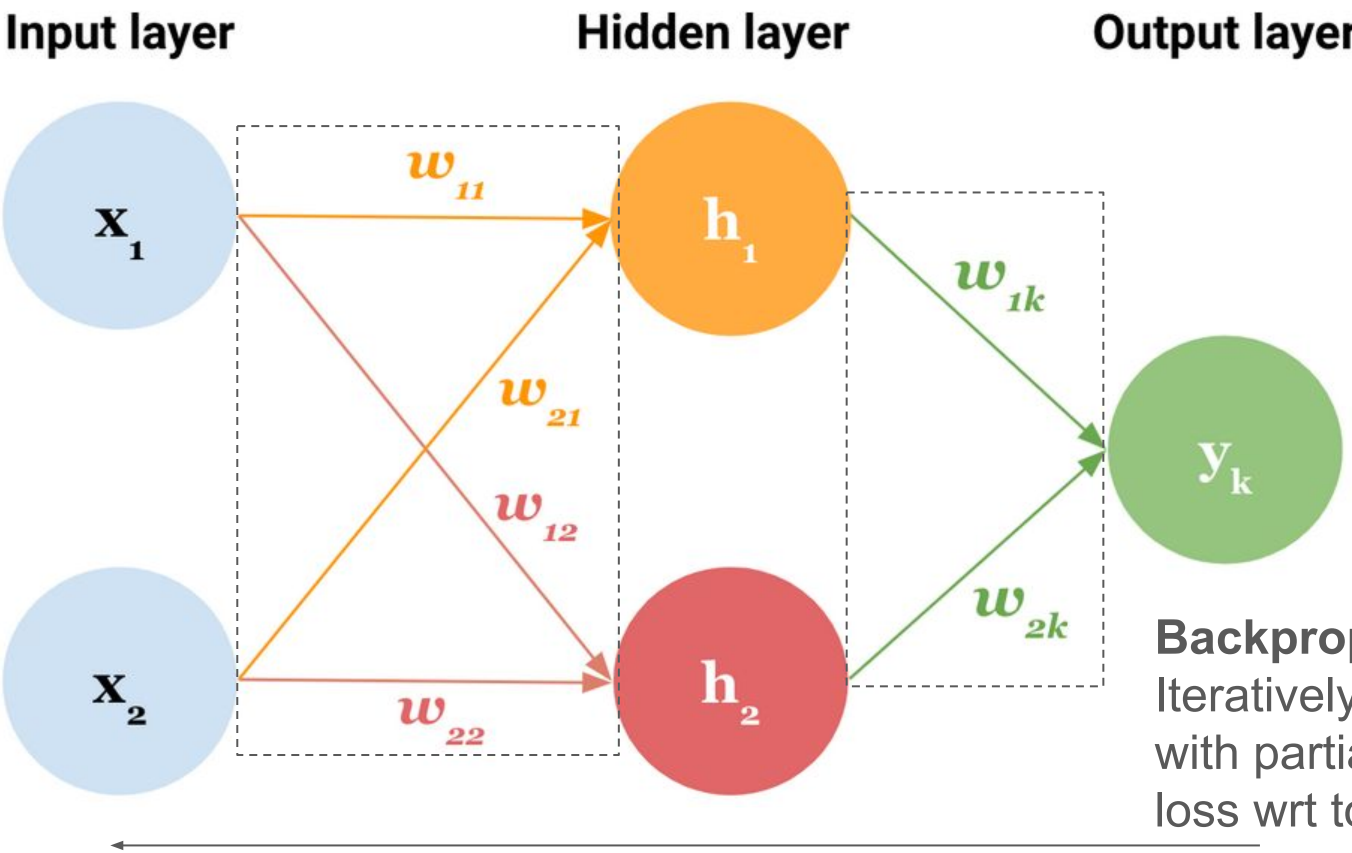


$$y_k = \frac{1}{1 + e^{-w_{1,k}h_1 + w_{2,k}h_2}}$$

Neural Networks

- Need: efficient models that capture high-dimensional non-linear relationships
- Solution: stack many simple models with a non-linearity (e.g., logistic / ReLU)
- **Neural Networks:**

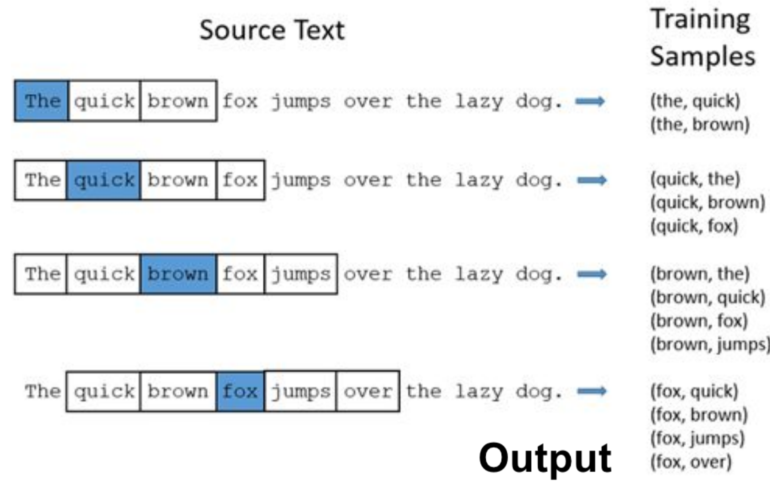
Efficient Training?

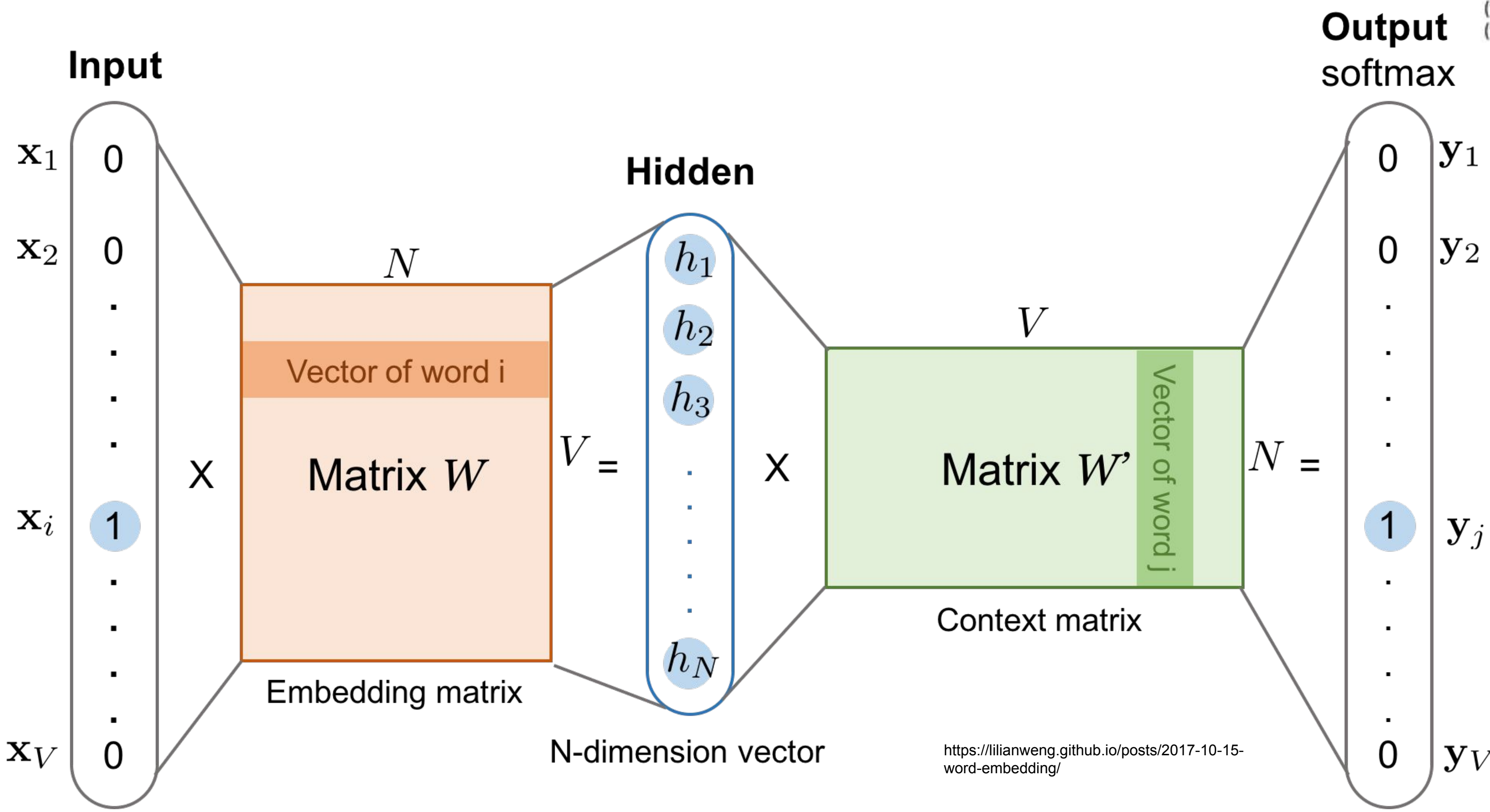


Output = ŷ
Target = y
Loss = 1/2*(y-ŷ)²

Backpropagation:
Iteratively update weights with partial derivative of loss wrt to weights

Learnt word embeddings

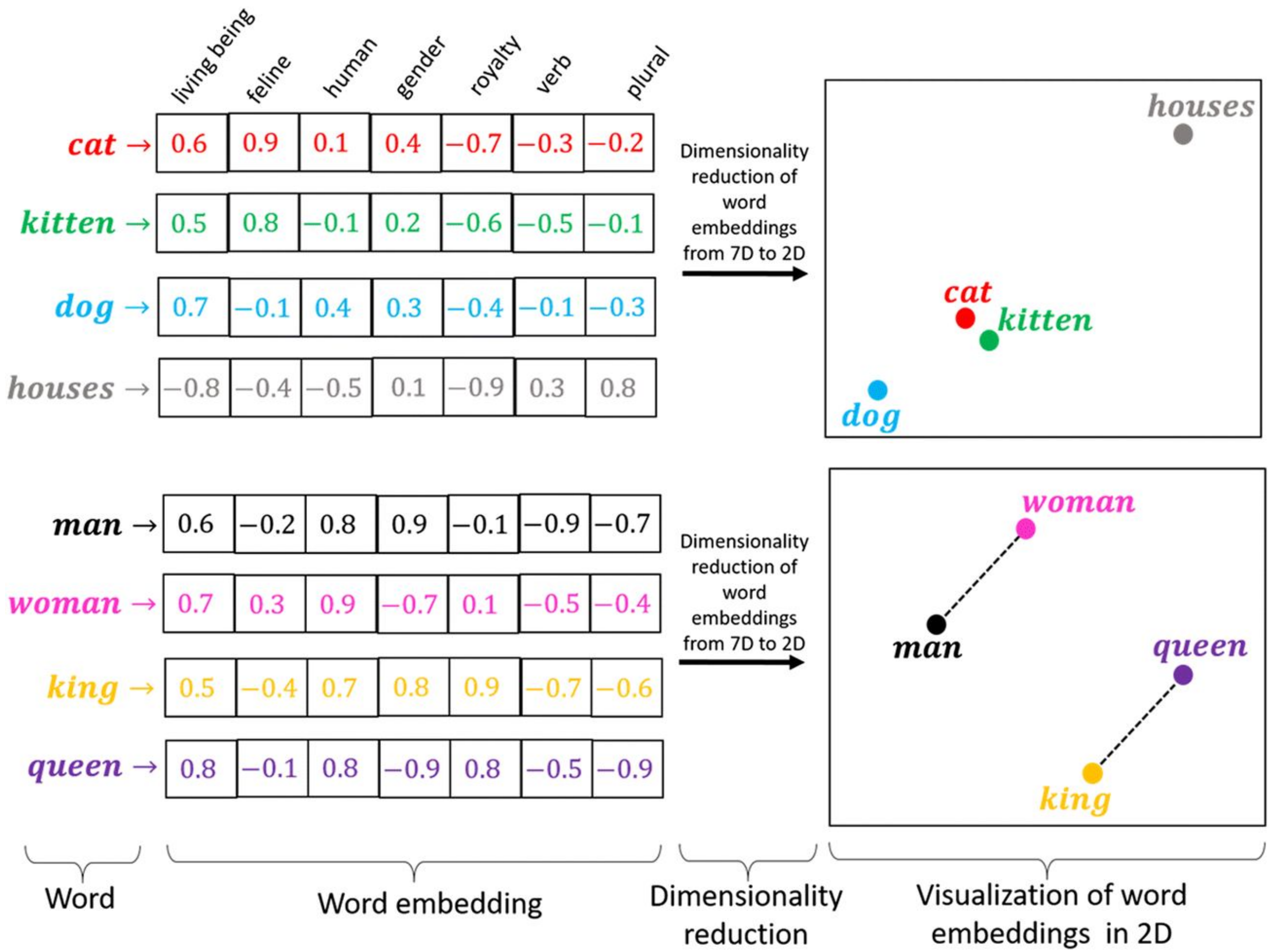




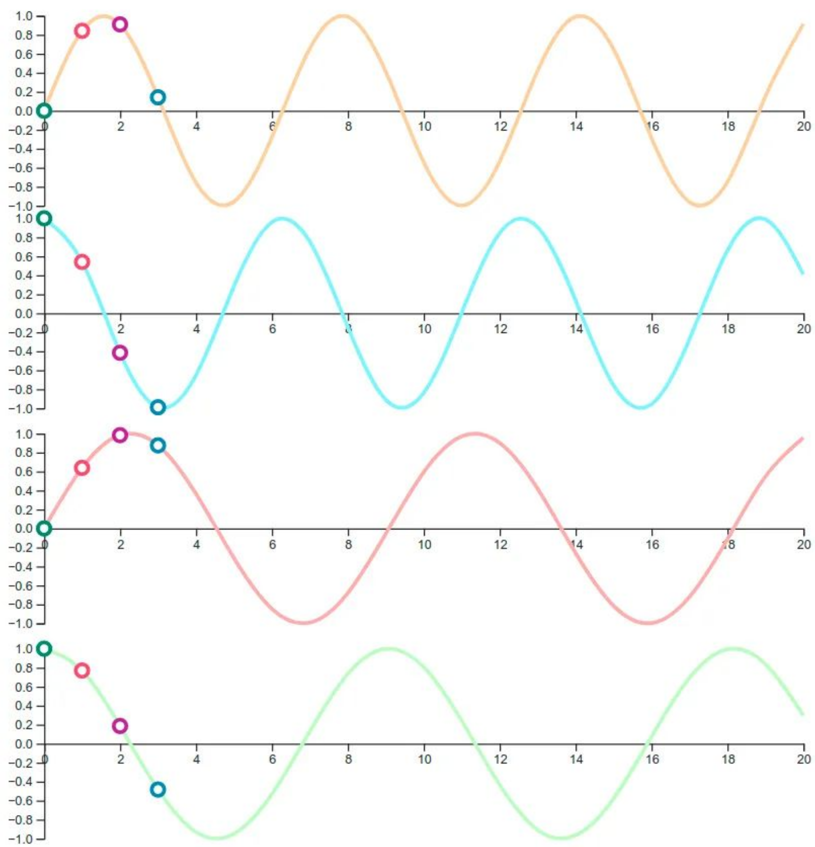
https://lilianweng.github.io/posts/2017-10-15-word-embedding/

Word2Vec/GloVe/FastText -> ELMo -> BERT -> ERNIE -> GPT-3/Megatron/T5 -> GPT-4/Llama3

Learnt embeddings are powerful



Beyond word embeddings: encoding position



p0	p1	p2	p3	
0.000	0.841	0.909	0.141	**i=0**
1.000	0.540	-0.416	-0.990	**i=1**
0.000	0.638	0.983	0.875	**i=2**
1.000	0.770	0.186	-0.484	**i=3**

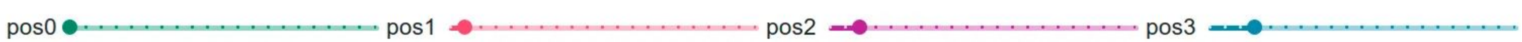
Positional Encoding

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

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

Settings: d = 50

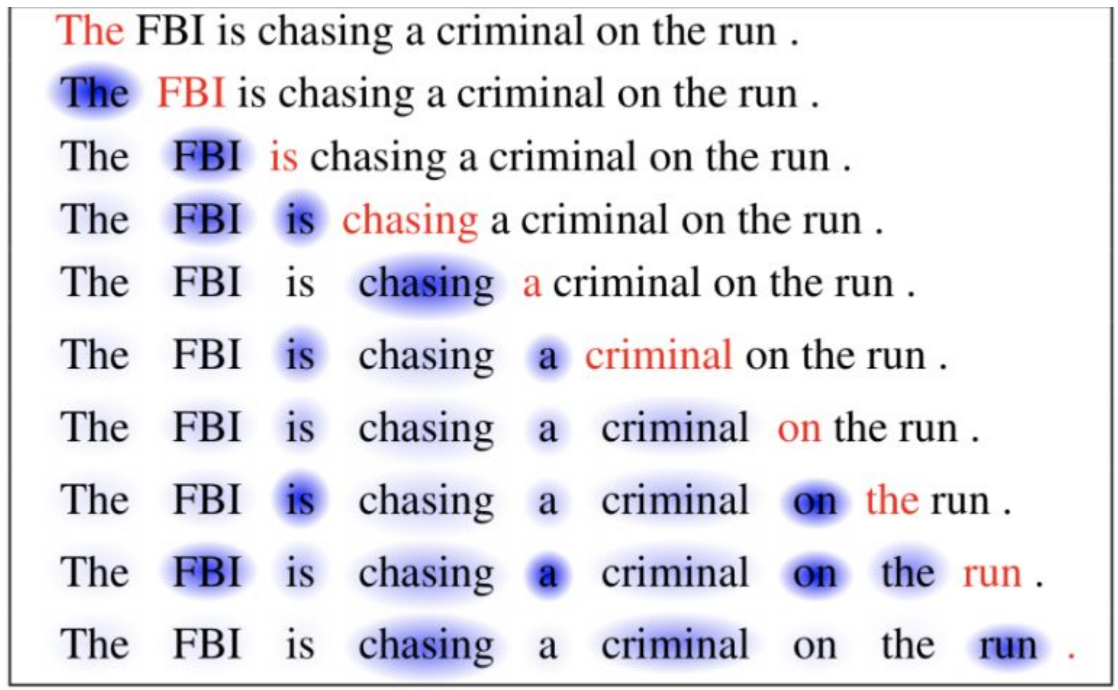
The value of each positional encoding depends on the *position* (*pos*) and *dimension* (*d*). We calculate result for every *index* (*i*) to get the whole vector.



https://shorturl.at/5GUAb

Attention mechanisms (massive topic!)

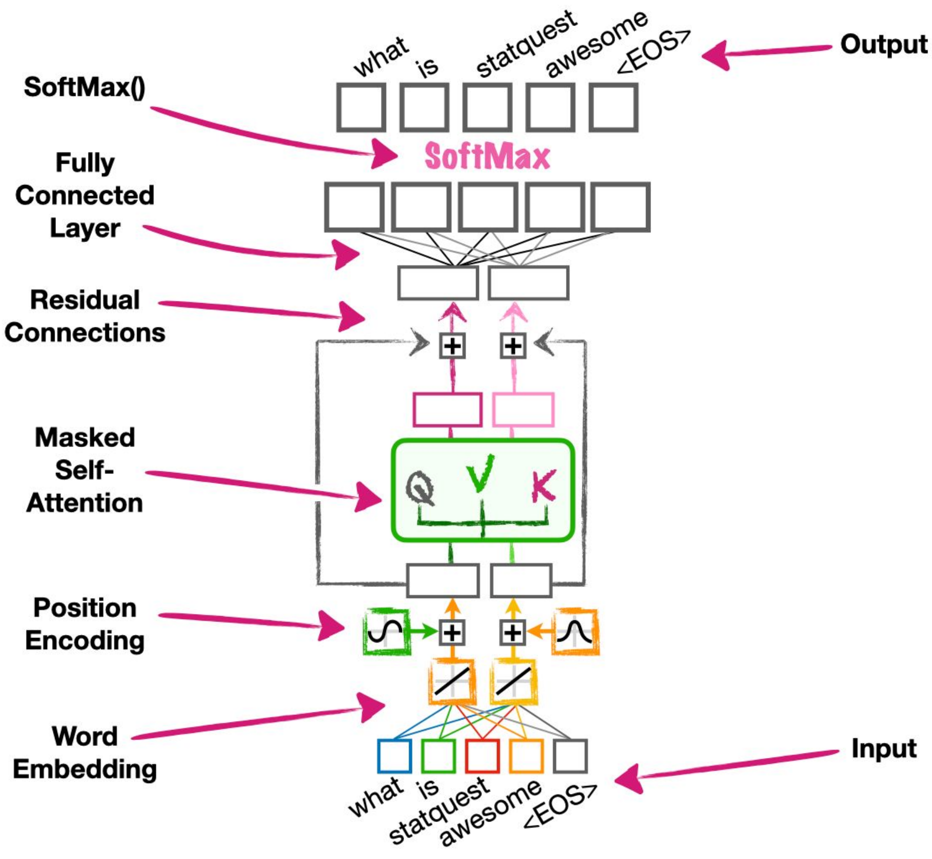
- Self-similarity vs similarity to other words
- Auto-regressive (mask self-attention) if only prior words



https://shorturl.at/KCfx1

Attention mechanisms (massive topic!)

- Self-similarity vs similarity to other words
- Auto-regressive (mask self-attention) if only prior words
- Combining all these mechanisms with a lot of data gives you transformer models (e.g., GPT1-4)

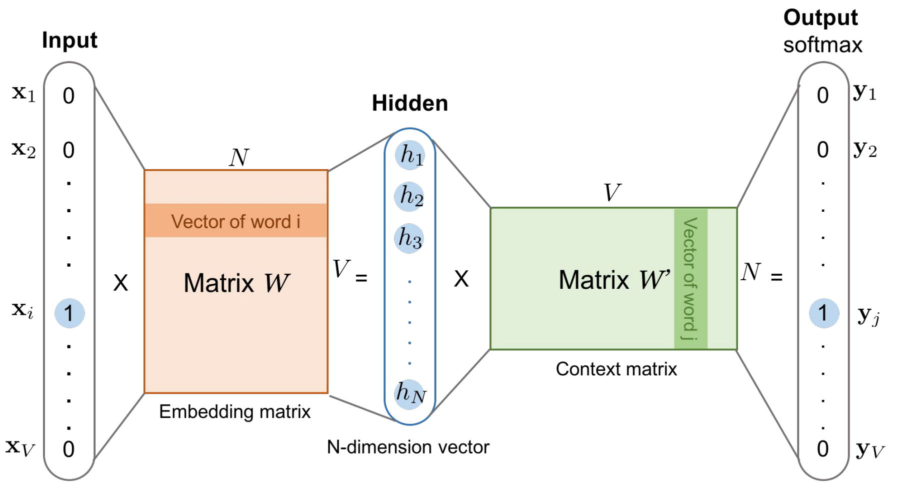


https://shorturl.at/J1ff6

Custom embeddings and fine-tuning

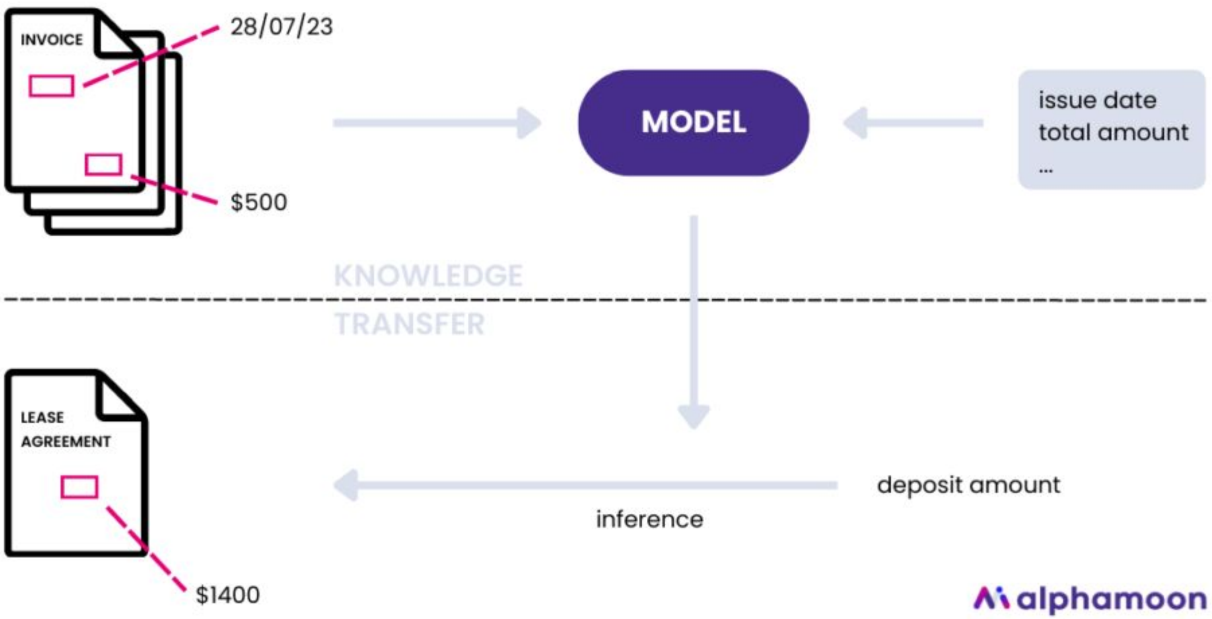
- Same approach can be used beyond just words:
- Med2Vec
- EHR2Vec
- BioALBERT

- Corpora used to create embedding may not be a good fit for specialised text (i.e., EMRs aren't representative of the internet at large… we hope).
- Repeat training on your data but initialise with pre-trained weights



Multimedia/multimodal embeddings

- This approach can be extended to joint embeddings of multiple data types (e.g., “multimodal” CLIP embeddings/Diffusion in Module 3)
- Zero-shot: using model trained on your unrelated problem with no fine-tuning

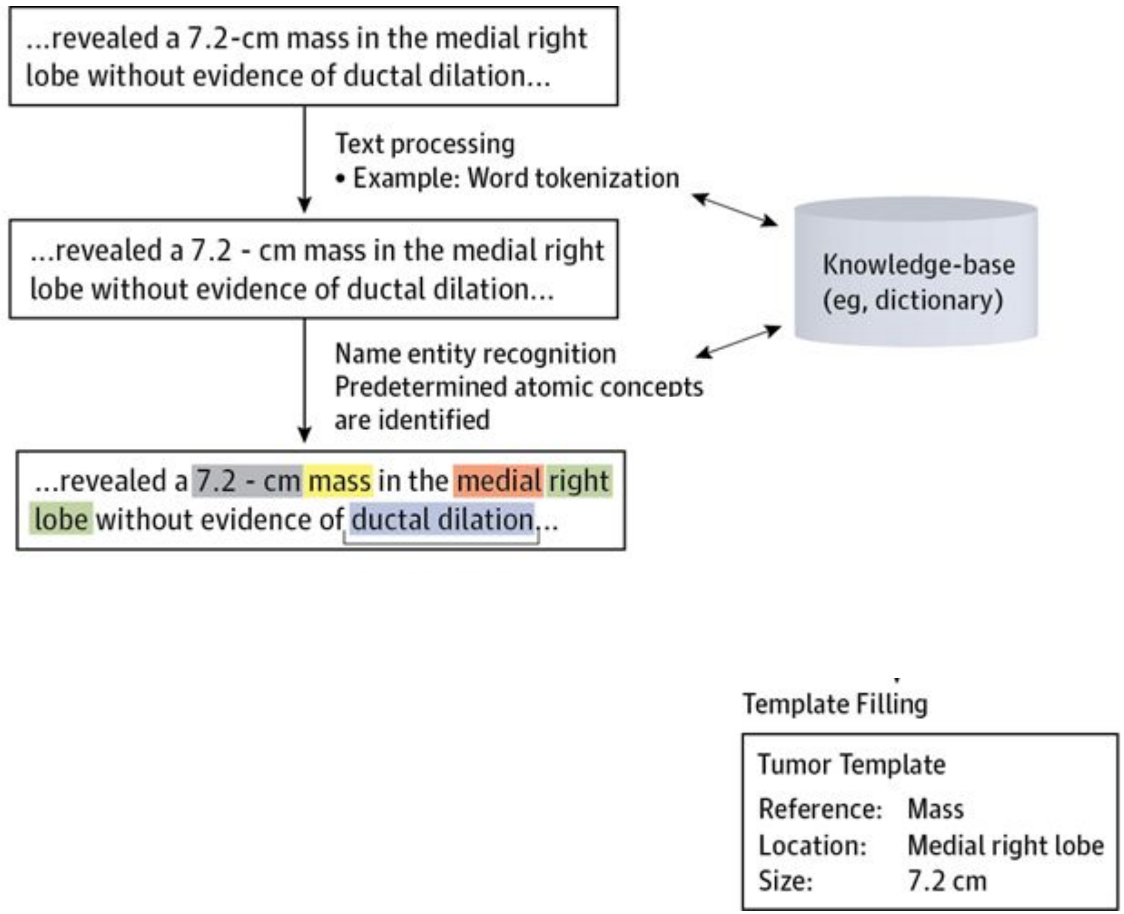


- May require exponential data (especially if multimodal)

https://arxiv.org/pdf/2404.04125

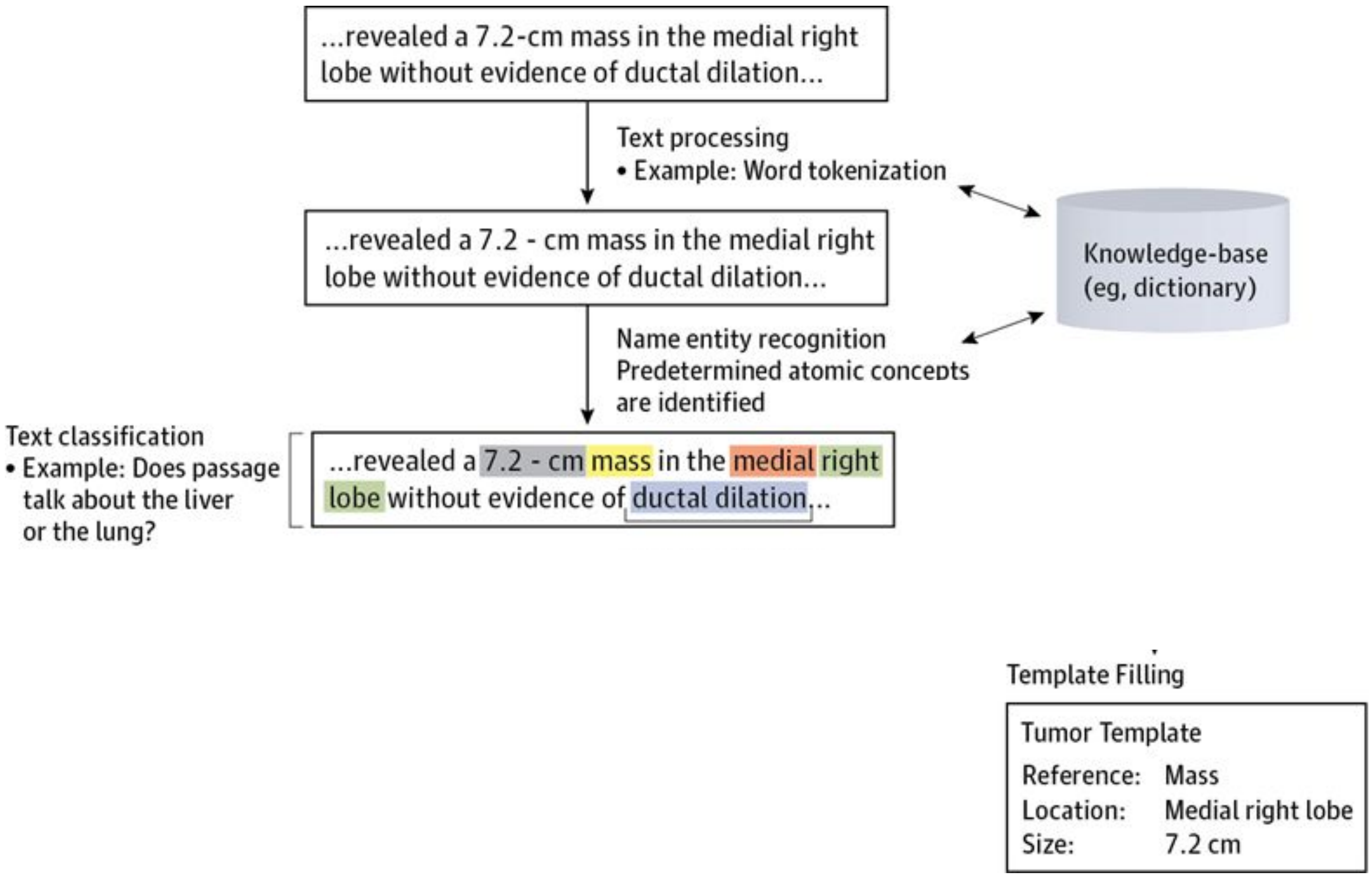
With embeddings we can build/use models for more complex problems

Train classifier on labelled medical text (e.g., ontology) = Named Entity Recognition



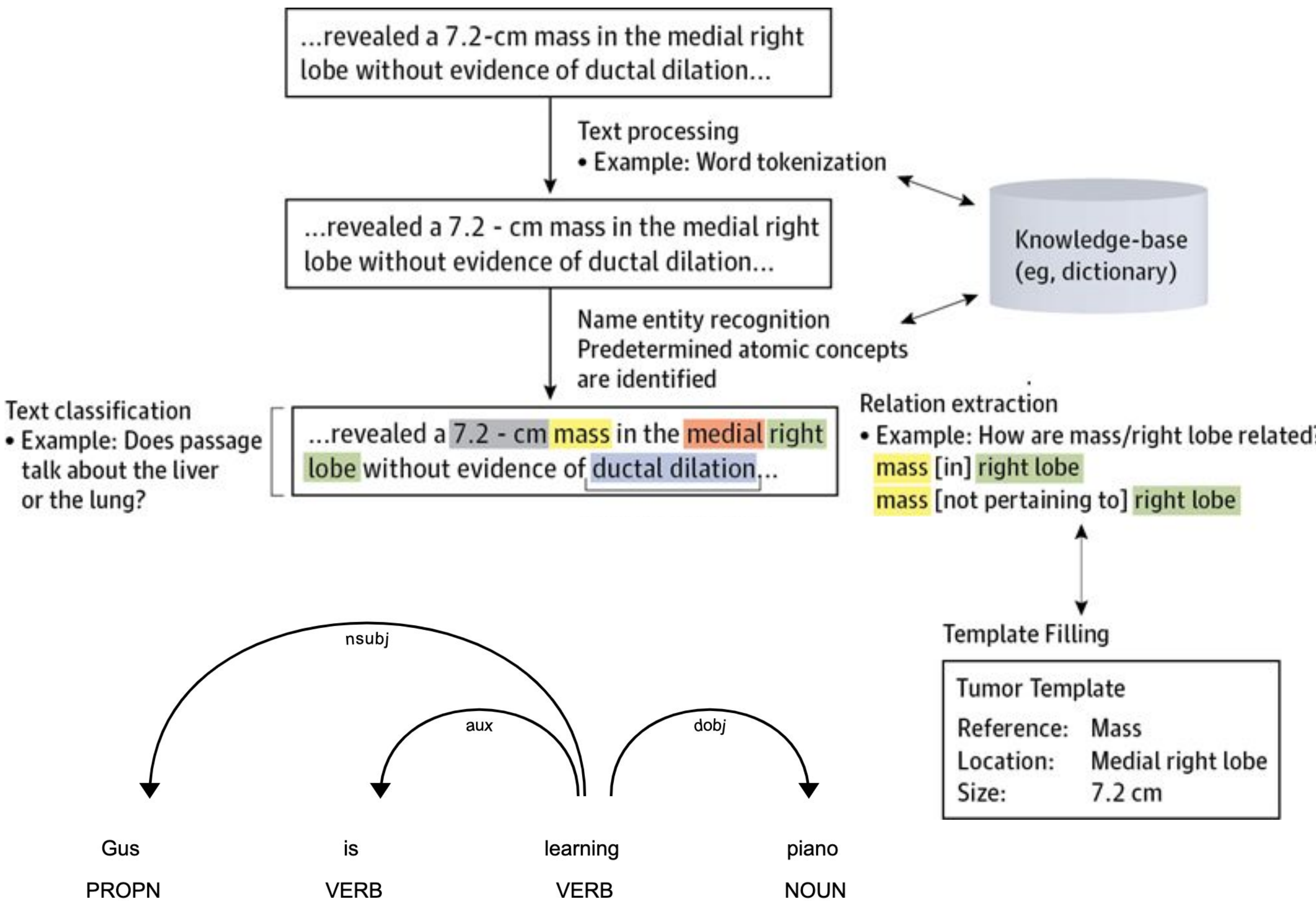
10.1001/jamaoncol.2016.0213

Train document classifier on EMR notes labelled by organ => Text classification



10.1001/jamaoncol.2016.0213

Use classifier trained to identify parts of speech and their relations (previously HMMs)



Overview

- Describe electronic medical/health record systems and the types of data they typically contain
- Distinguish structured, semi-structured, unstructured text data
- Describe approaches to searching text
- Outline key steps in preparing text for analysis
- Explain the general concept of learnt word embeddings
- Explain how embeddings can be tuned/customised
- Identify differences between named entity recognition, parts of speech tagging, and dependency parsing

- ***Not covered: fuzzy search and text indexing***