NATURAL LANGUAGE PROCESSING FOR READING PAPERS

Logan Ward
Asst. Computational Scientist
Argonne National Laboratory

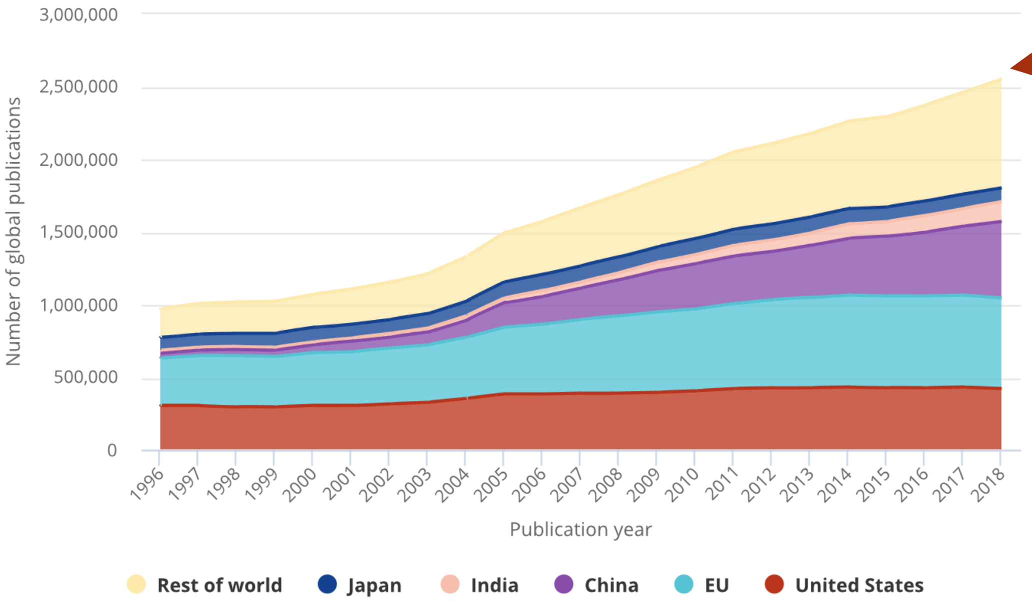
6 March 2022

Scientific literature is a deluge

National Science Board | *Science & Engineering Indicators* | NSB-2020-6

FIGURE 5A-2

S&E articles in all fields, for selected regions, countries, and economies and rest of world: 1996–2018



EU = European Union.

Note(s)

Article counts refer to publications from a selection of peer-reviewed journals and conference proceedings in S&E fields from Scopus. Articles are classified by their year of publication and are assigned to a region, country, or economy on the basis of the institutional address(es) of the author(s) listed in the article. Articles are credited on a fractional-count basis (i.e., for articles produced by authors from different countries, each country receives fractional credit on the basis of the proportion of its participating authors). Data are not directly comparable to *Science and Engineering Indicators 2018*; see Technical Appendix for information on data filters. For more information on the 2019 World Bank Country and Lending Groups classification of income groups, see https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups, accessed January 2019. Data by country are available in Table S5a-2.

Source(s)

National Center for Science and Engineering Statistics, National Science Foundation; Science-Metrix; Elsevier, Scopus abstract and citation database, accessed June 2019.

Science and Engineering Indicators

2.5M science and engineering articles per year!

My Mendeley has 1086 articles, and I would be generous to claim I had mastery of half of them

Computers are needed to manage this, but how!?

Step 1: Define tasks (of roughly greater complexity)

Semantic Search: Being able to search based on meaning

Information Extraction: Get data out of text

Summarization: Creating summarizes of certain text

Question Answering: Generating responses to queries

Logical Reasoning: Identifying conflicting data, testing theories

BASICS OF NLP FOR INFORMATION EXTRACTION

Understanding “natural” text is hard.

What tasks go into information extraction?

High-density polyethylene has a glass-transition temperature of –110 C.

"Named Entity Recognition"

- “polyethylene“ is a polymer
- “glass-transition temperature" is a property

"Relationship Extraction"

- "polyethylene" has a characteristic "high-density"
- "glass-transition temperature" has a value "-110"
- "-110" has units of "C"
- "polyethylene" has a property “glass-transition temperature”

```
{
  "material": "polyethylene",
  "processing": {
    "density": "high
  },
  "properties": {
    "t_g": {
      "value": -110.,
      "units": "C"
    }
  }
}
```

Let's consider one problem: Named-Entity Recognition

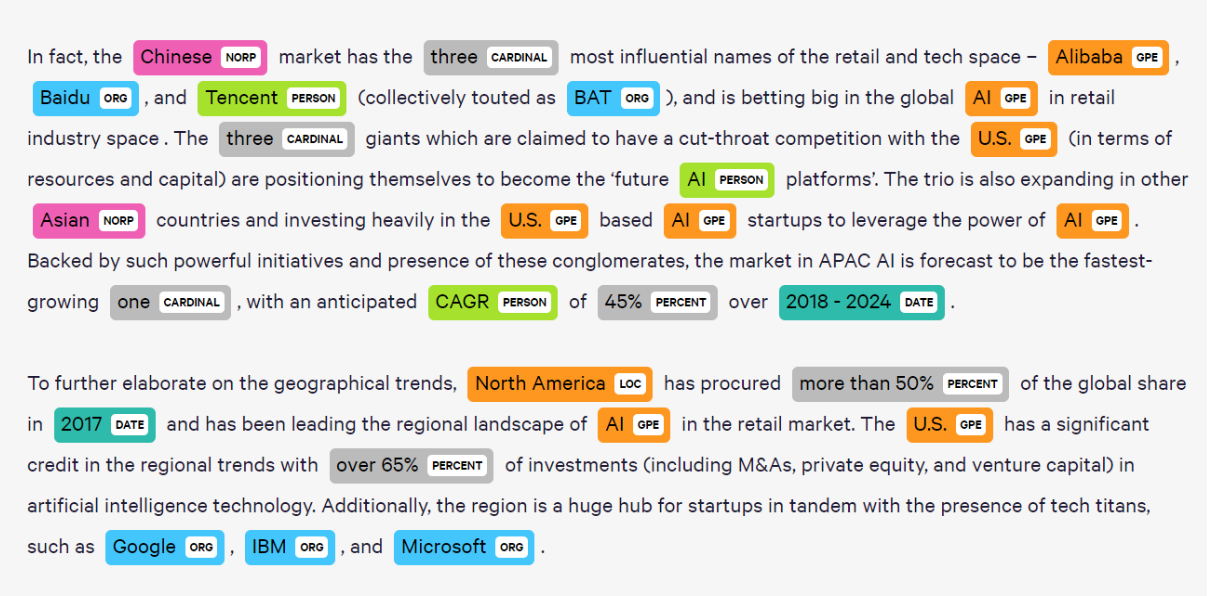
Named Entity Recognition (NER) is a supervised learning problem

Inputs: A word [and maybe its context]

Output: Category classifications

Examples:

- Is Apple a noun?
- Is that noun a person, place or thing?



Solvable with machine learning, if would could turn a word into "features..."

Source: Medium blog

Word embeddings are features

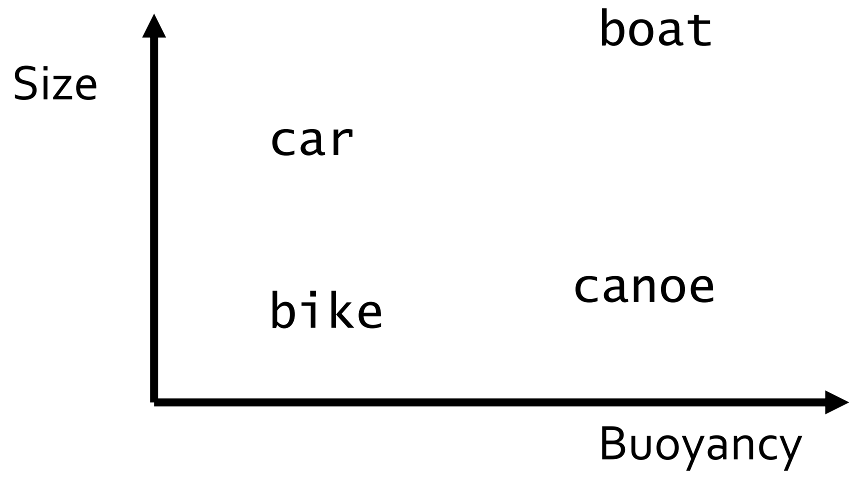
Idea 1: One input per word

```
canoe     =     [1, 0, 0, 0]
ship      =     [0, 1, 0, 0]
bike      =     [0, 0, 1, 0]
car       =     [0, 0, 0, 1]
```

Problem: Information lean!

- $>10^5$ features for some languages
- Mutually orthogonal for each word

Idea 2: *Embed* words with meaning



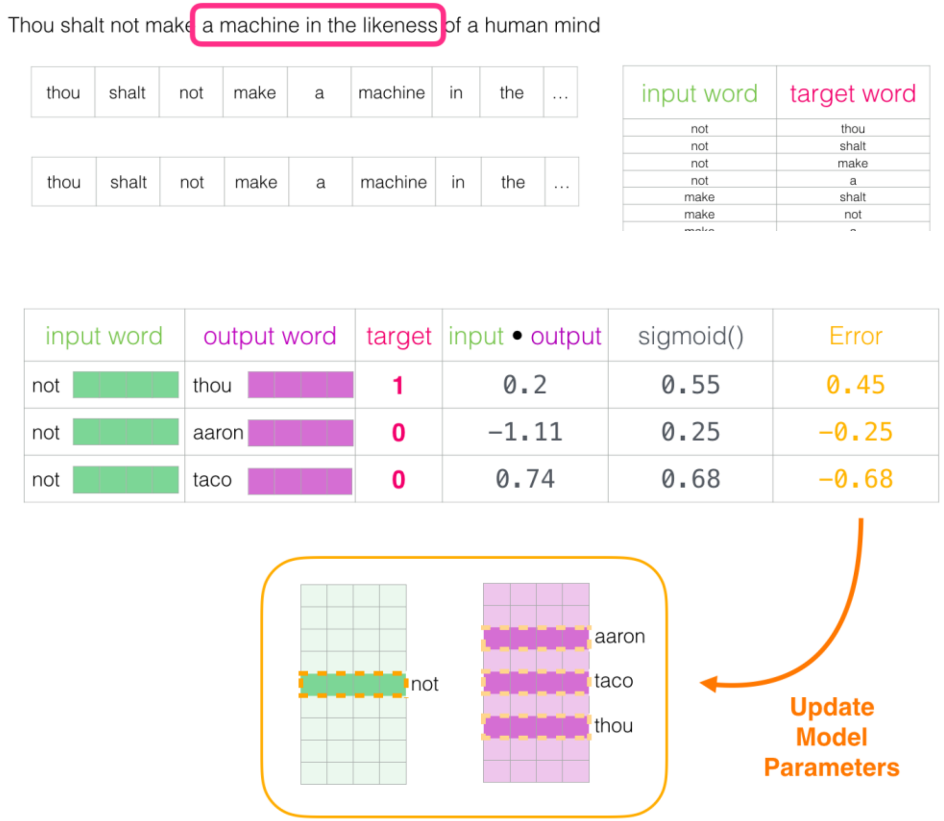
Problems are fixed!

- Arbitrary number of features
- Feature vectors encode meaning

Embeddings are from "unsupervised learning"

General concept: Related words have similar contexts

<u>**One**</u> **way to use this concept?** Learn a word-prediction model with "skipgram"



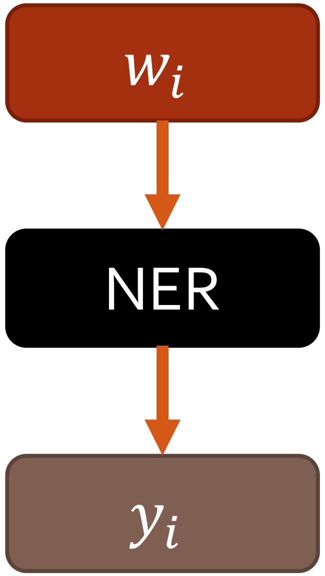
1. Get contexts of words in text (positives)
2. Get random pairs of unrelated words (negatives)
3. Assign each word a random embedding
4. Iteratively update the embeddings

Source: Jay Alammar's excellent blog (and great taste in Sci-Fi); d2l.ai/chapter_natural-language-processing

Learning a NER model

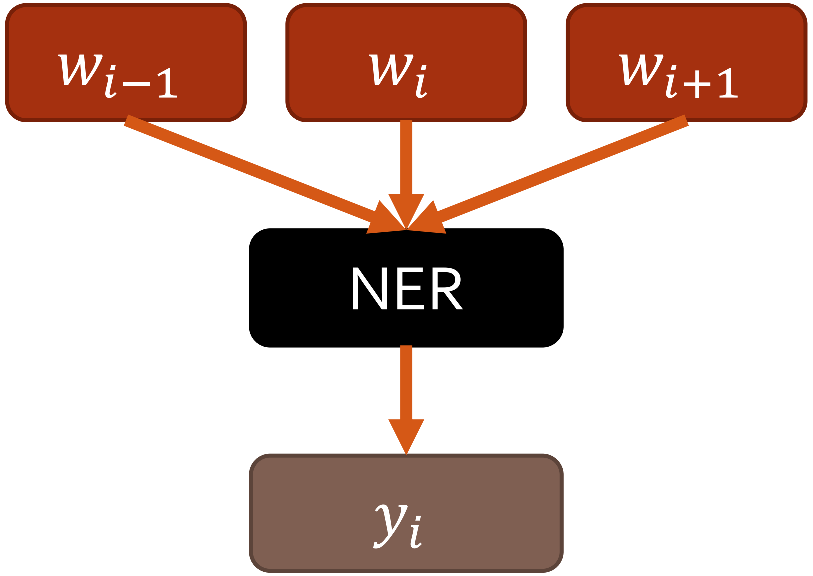
Smart

Use a single word



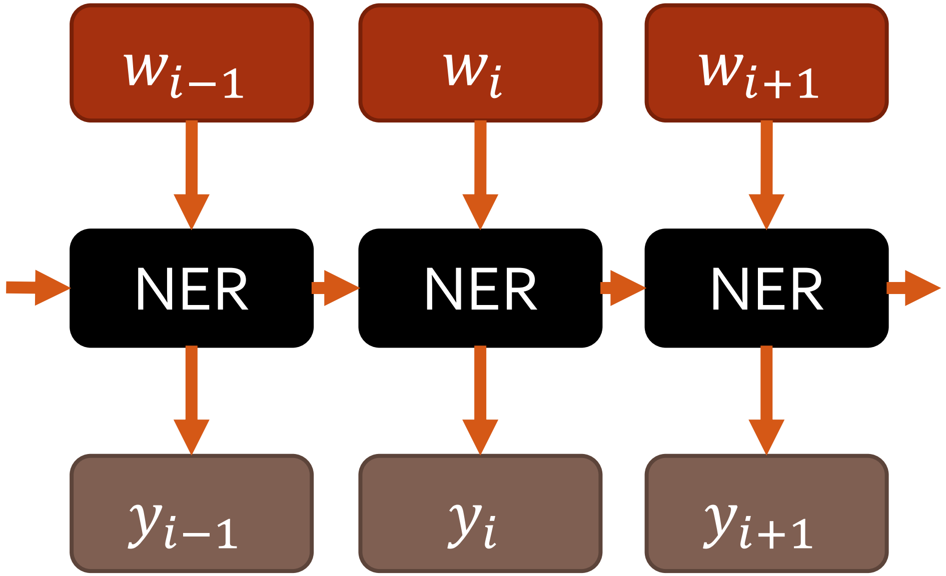
Smarter

Use its context



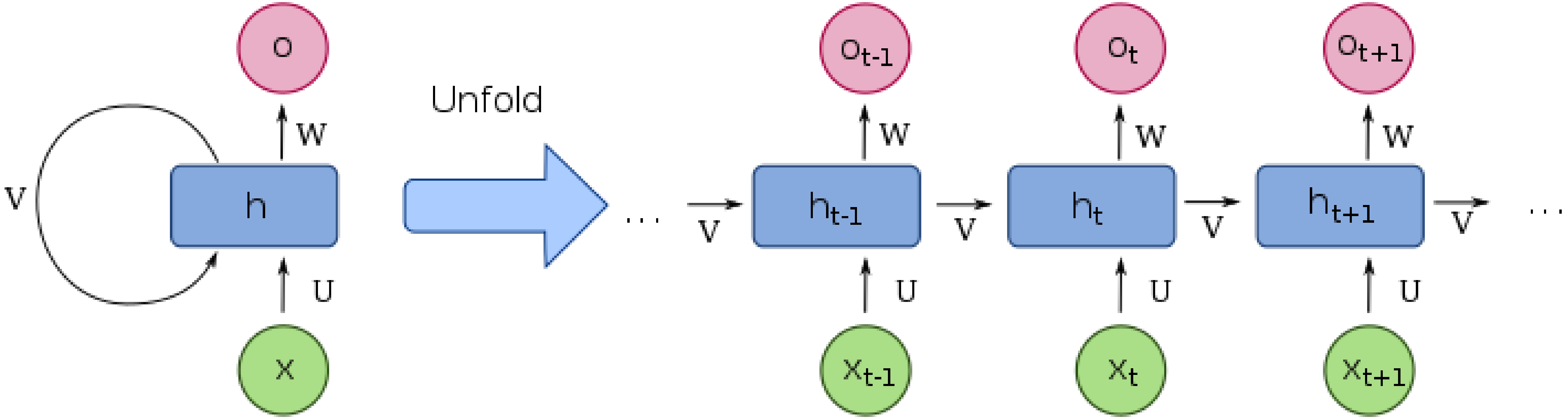
Smartest!

Use the whole sentence



Recurrent neural networks (RNNs)

RNNs learn how a hidden state "h" evolves with a sequence of data (x)



Composed of 3 learnable functions (terms are mine):

U – Update function: maps data, x_i, from a sequence to add to a state, h_i

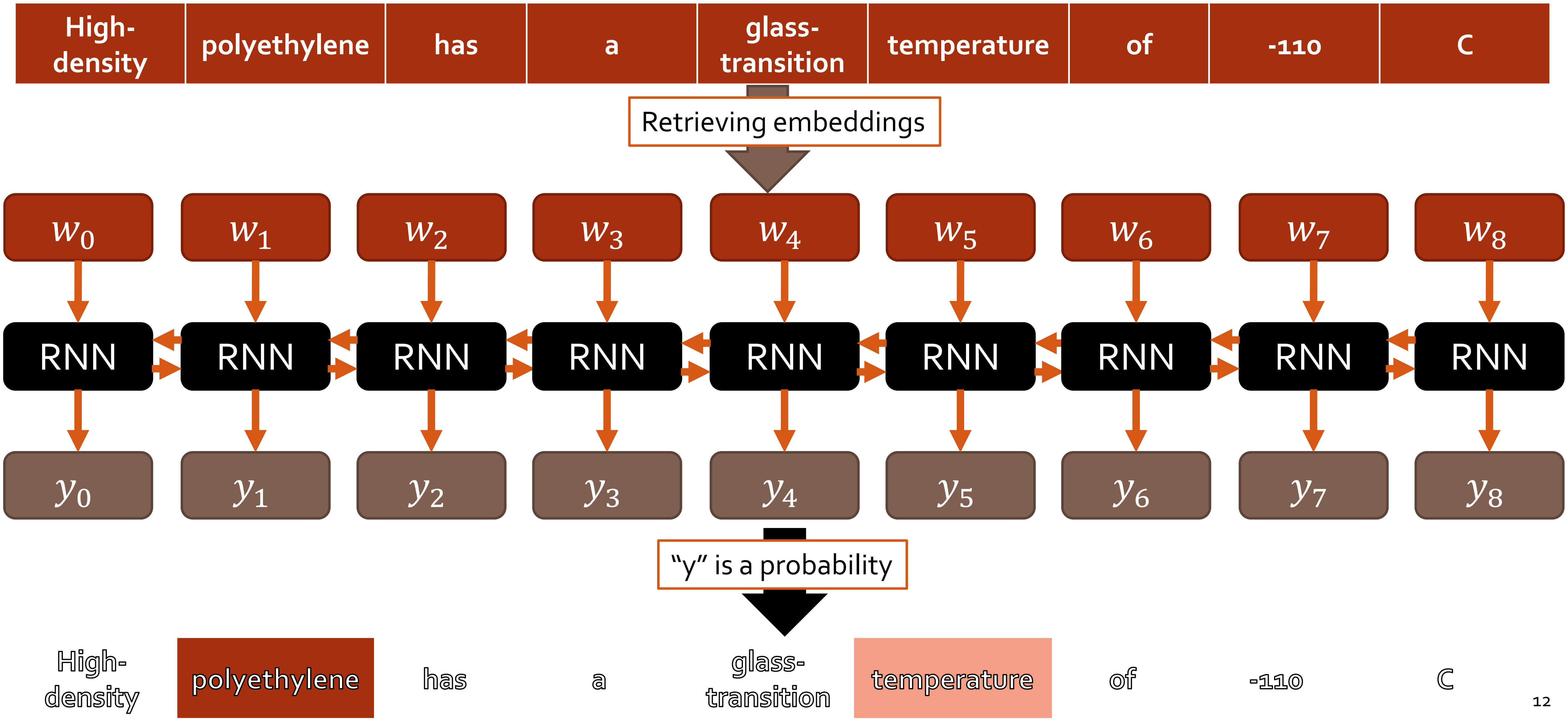
V – Propagation function: updates state, h_{i-1}, to account for change over time

W – Output Function: maps current state to an output

Many ways to build these functions: GRU, LSTM

Source: Wikipedia

Example for NER and “bi-directional” RNN



Avoiding out-of-vocabulary words

You can do "character-level" versions of RNNs as well

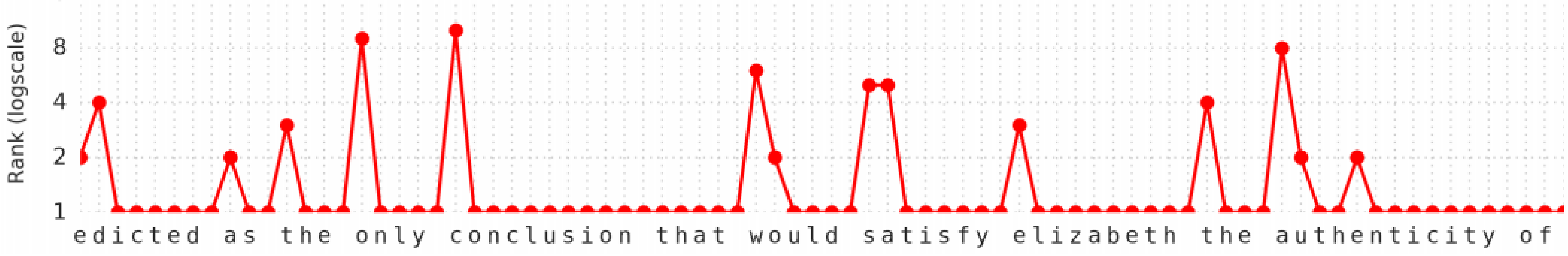


Figure 6: Per-character entropy, loss and rank assigned by T64 after seeding on the 512 character sequence from Figure 5.

Require bigger models and more training data, but offer great flexibility

Source: Al-Rfou et al. AAAI (2019)

Last word: There are super-well-used codes for NLP

Codes for processing language data

spaCy

NLTK

Ways to label data easily

prodigy

Pretrained language models

GPT-3 Model Card

Last updated: September 2020

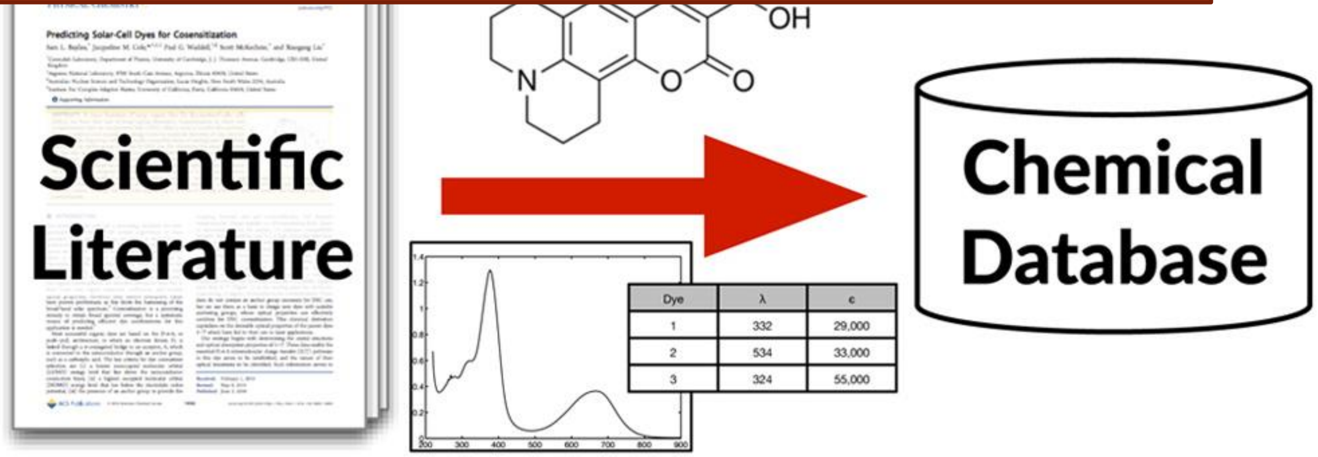
Inspired by Model Cards for Model Reporting (Mitchell et al.), we're providing some accompanying information about the 175 billion parameter GPT-3 model.

Google twice before trying to code something yourself

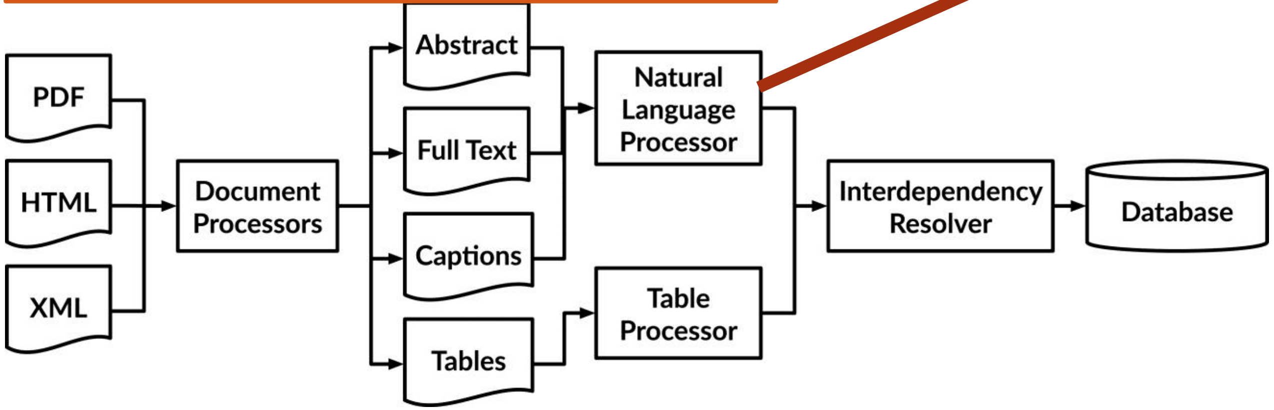
RECENT HISTORY OF NLP IN MATERIALS ENGINEERING

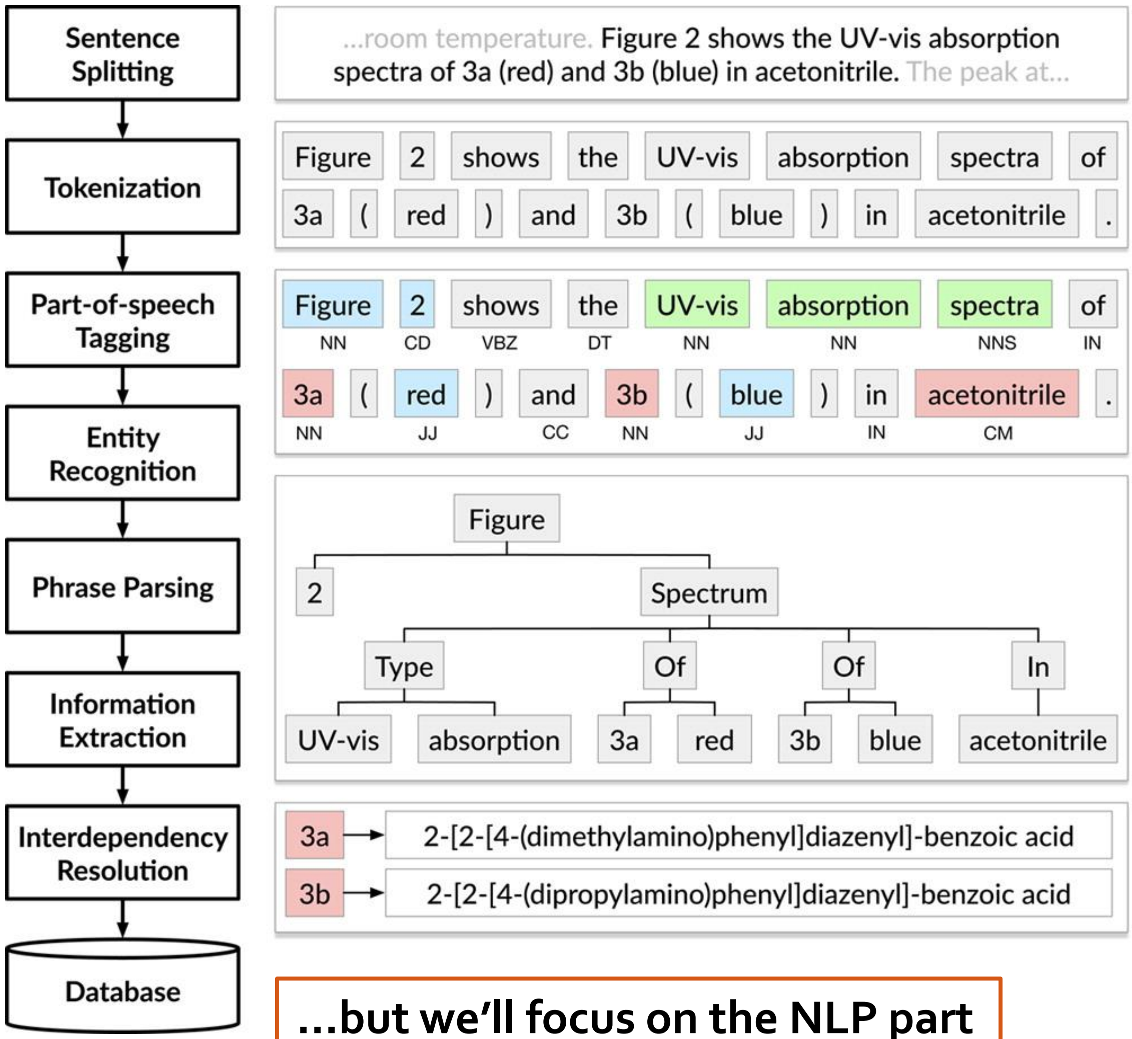
ChemDataExtractor (Swain and Cole, 2016)

Many moving parts to getting chemical data from papers



Parsing documents is a challenge…





…but we'll focus on the NLP part

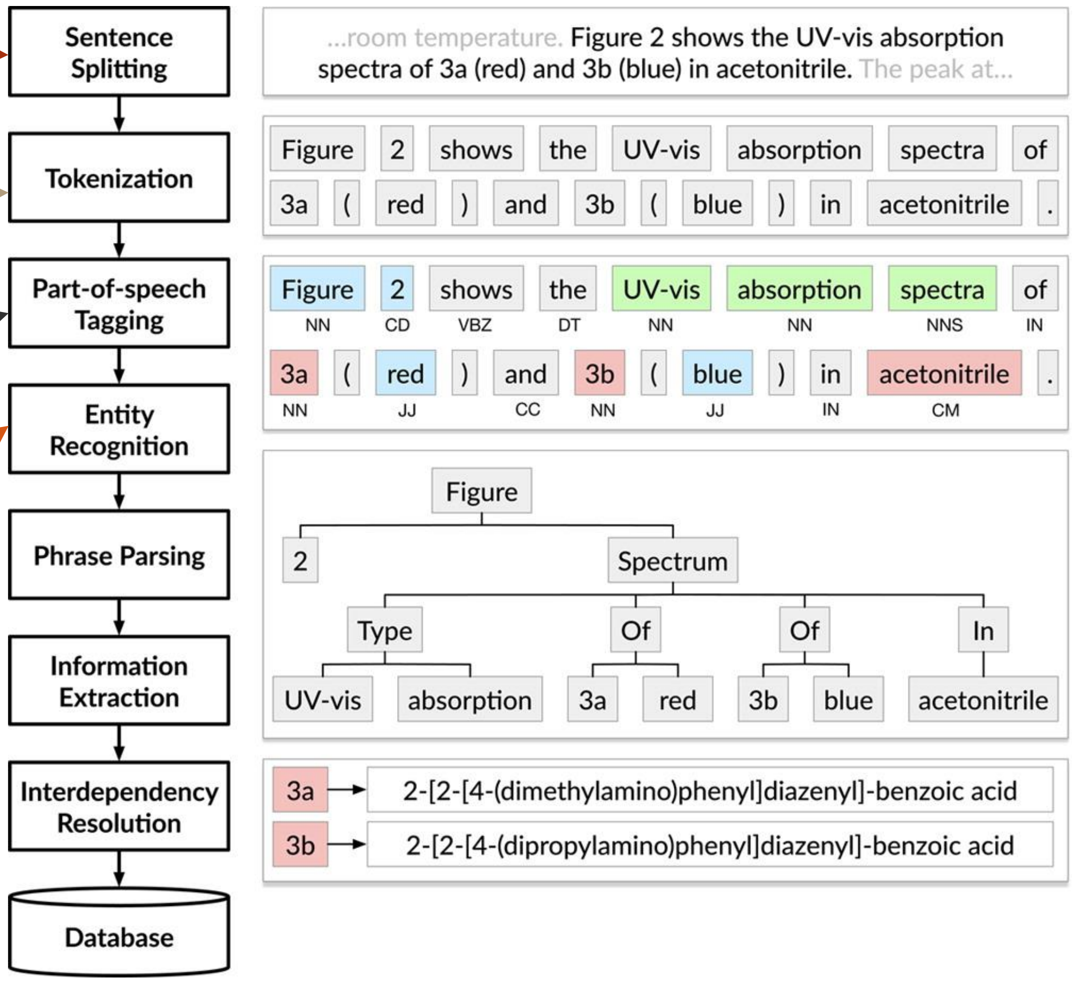
NLP required a lot of thought in many different steps

Does “et al.” mark the end of a sentence?

Is “poly(ethylene glycol)” 1 or three words?

Do I train a POS tagger on newspapers or MEDLINE abstracts?

Do I train a NER on newspapers or MEDLINE abstracts?



Source: Cole and Swain. JCIM (2016)

CHEMDNER: Labeled abstracts for chemical NER

Labeled dataset of 10k abstracts

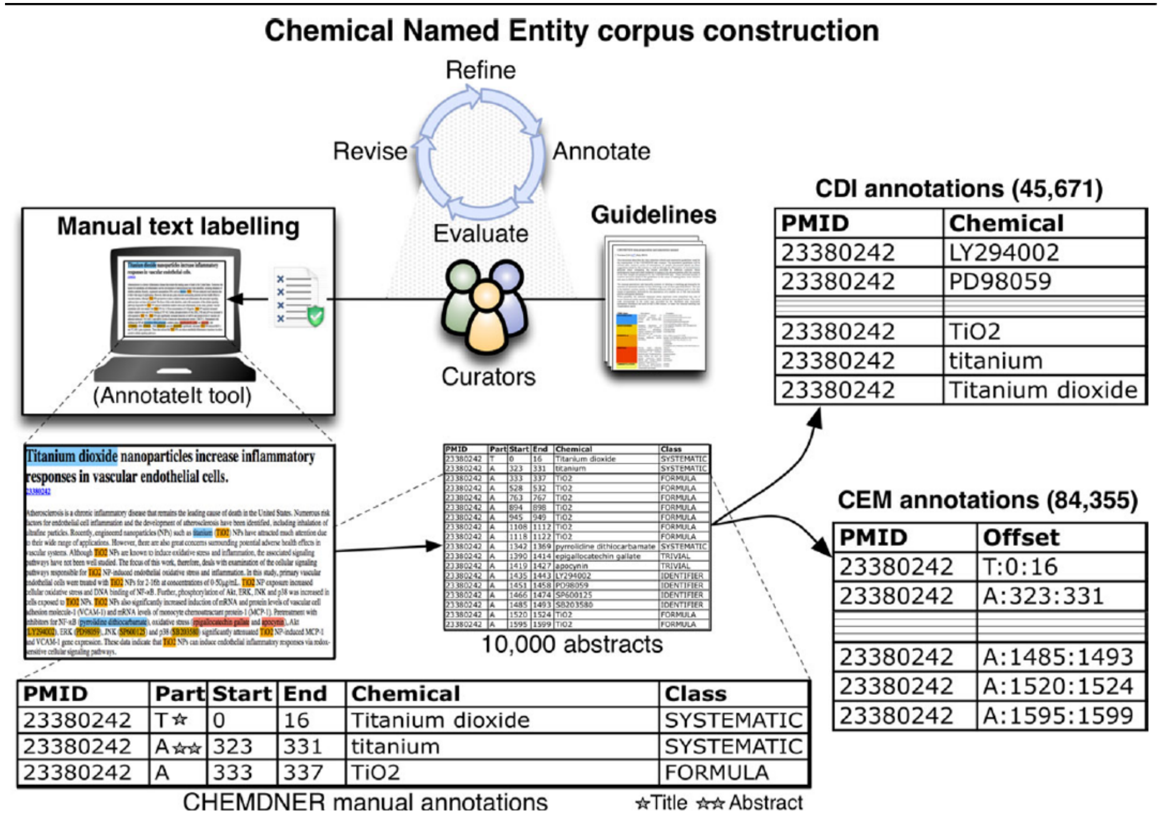


Figure 2 Left side: Overview of the manual CHEMDNER corpus annotation process. Right side and bottom: Annotation examples for the Chemical Document Indexing (CDI) and Chemical Entity Mention (CEM) task.

Used to create dozens of ML engines

CEM team rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Techniques																										
Machine learning	*	*		*	*	*	*	*		*			*		*	*	*	*	*	*	*	*		*	*	*
CRFs	*	*		*	*	*	*	*		*			*		*	*	*	*	*	*	*	*		*	*	
SVMs	*						*			*					*				*		*					
Logistic Regression																			*							
Max. Entropy													*													
Random Forests																				*						
Rule-based	*		*						*							*										*
Dictionary lookup only											*	*											*			
NLP																										
Tokenization	*			*	*	*	*	*	*	*	*	*	*		*	*	*	*		*	*		*	*	*	*
Suffixes	*			*	*		*	*	*	*	*			*	*		*	*	*	*	*			*		*
Sentence splitting	*			*	*	*	*	*	*	*	*	*	*		*			*			*				*	*
Named entities	*	*	*	*	*				*		*	*		*		*					*			*	*	*
Affixes	*			*	*		*	*	*	*	*				*			*		*	*			*		*
Word morphology	*			*	*	*	*		*	*					*		*				*			*		
POS tagger	*			*	*	*	*	*	*	*		*				*					*					*
Nomenclature rules	*		*	*	*				*		*			*							*			*		
Bigram, Trigrams, etc	*			*	*		*	*					*		*		*		*			*				
Lemmatization	*			*	*	*	*	*								*			*							
Stemming					*		*						*		*	*		*		*	*					
Shallow parsing						*			*																	
Bio-syllables					*																					
Deep parsing																					*					
Dictionary lookup																										
RegEx	*	*			*	*	*		*		*		*					*			*			*		
Rule-based variations	*			*							*			*		*			*							*
Suffix tree indexing		*																								*
Prefix trie lookups											*							*								
N-gram-based ASM																										*
Other			*							*										*		*	*		*	
Postprocessing																										
Filtering rules	*	*	*	*		*	*		*		*		*					*	*		*	*	*	*	*	*
Stop words			*			*			*		*	*		*		*		*			*		*	*	*	*
English dictionary																							*			
Filtering other entities																										*
Other					*																*					

Source: Krallinger et al. JCI (2015)

ChemDataExtractor: ML + Dictionary for NER

Table 2. Features Used in CRF Chemical Named Entity Recognizer[a]

CRF is a sequence/graph-based ML technique

feature	context	description
word	$w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$	normalized lowercase token text
POS tags	$w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$	part-of-speech tag
word shape	$w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$	simplified token representation
Brown clusters	$w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$	4, 6, 10, and 20 bit binary path prefixes
length	w_i	number of characters in token
counts	w_i	digit, upper, and lower case letter counts
prefixes	w_i	1−5 character prefixes
suffixes	w_i	1−5 character suffixes
hyphenated	w_i	contains a hyphen character
alphabetical	w_i	contains only alphabetical characters
case	w_i	upper, lower, or title cased
number	w_i	number in digit or word form
punctuation	w_i	contains only punctuation characters
URL	w_i	looks like a URL

[a] A context window is used, such that some features for the token at index i are derived from the token text (w) of surrounding tokens.

An embedding, like word2vec

Character-level features about words

Better performance by adding rules/dictionaries

Table 8. Precision, Recall, and F-Score of Conditional Random Field (CRF), Dictionary, and Regular Expression Chemical Named Entity Recognizers When Used Separately and in Combination

system	precision	recall	F-score
CRF	90.5%	80.0%	84.9%
dictionary	88.6%	70.2%	78.3%
regular expression	89.4%	11.0%	19.6%
combined system	89.1%	86.6%	87.8%

Source: Cole and Swain. JCIM (2016)

Rule-based association engine

Associations between entities are assigned with rules

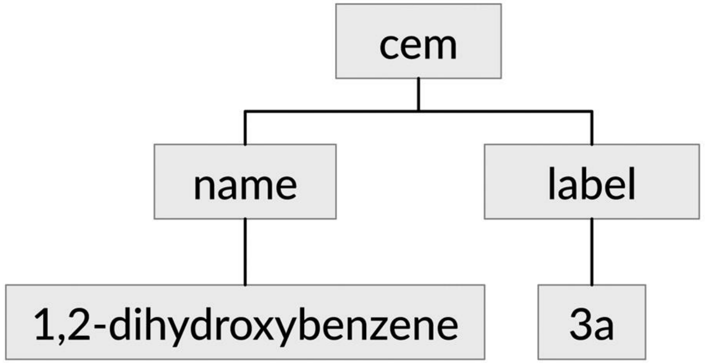
a) Chemical identifier grammar

```
name = T('B-CM') + ZeroOrMore(T('I-CM'))

label = R('[1-9][a-z]?') | R('[IVX]+')

cem: name + W('(') + label + W(')')
```

b) Parse tree for " … 1,2-dihydroxybenzene (3a) … "



Source: Cole and Swain. JCIM (2016)

The Synthesis Project

Full stack of NLP tools: Word embeddings, NER tools, labeled datasets!

Tour de force of using text for ML for materials and shining example of openness

Source: https://www.synthesisproject.org/

MatScholar: Disovery using word embeddings

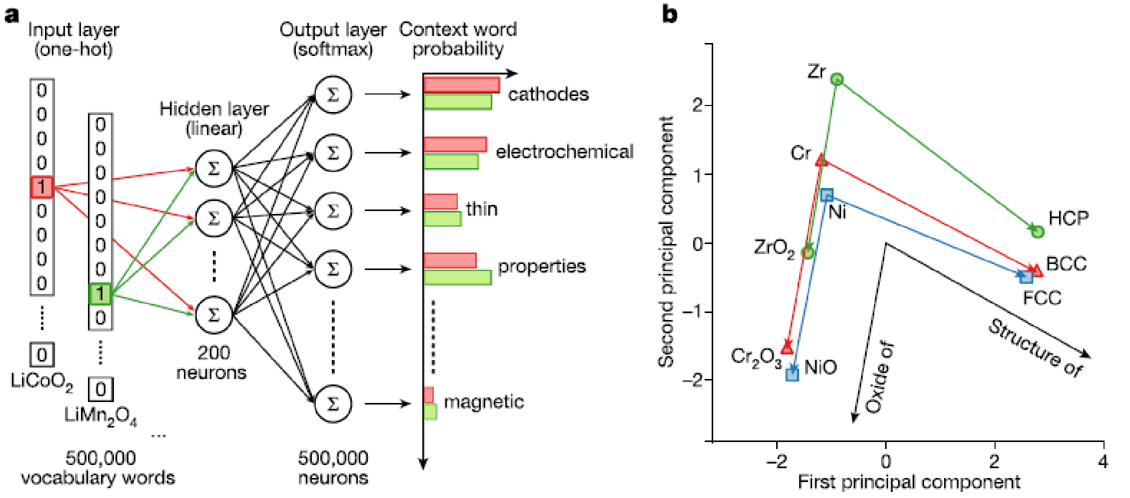
Encoding knowledge of materials space using word embeddings

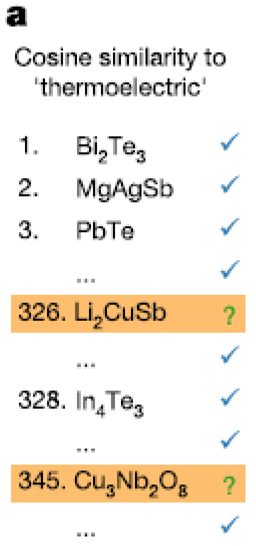
✓ No labelling required

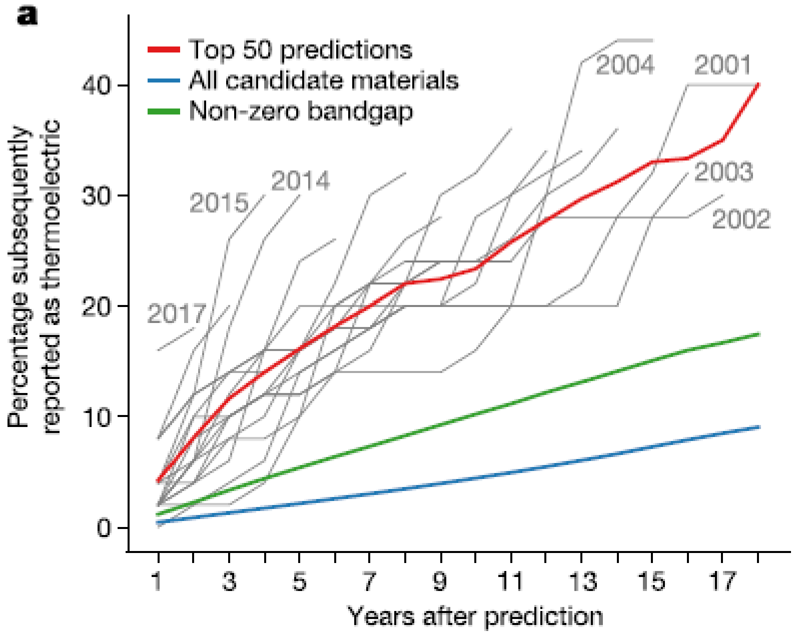
✓ Just learning how materials are described

Embeddings can predict good materials.

Of the Top 50 materials similar to "thermoelectric" based on pre-2000 papers, 40% have been discovered to be thermoelectrics







Source: [Tshitoyan et al. Nature (2019)]()

AVOIDING NLP

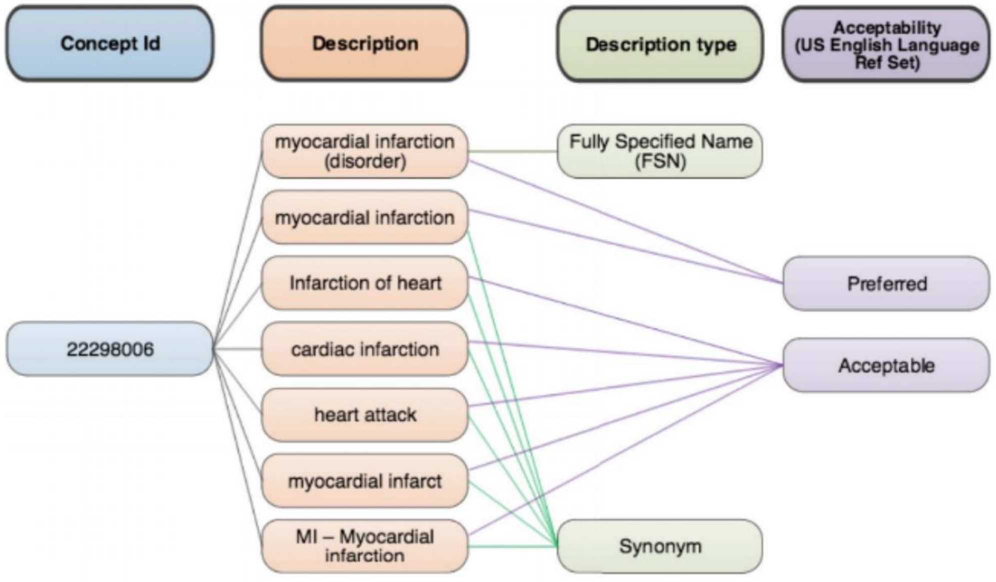
Humans can make text more understandable
if they just wrote better!

Ontologies/Controlled Vocabularies

Some communities enforce precision in language

SNOMED CT: Definition of medical terms and mappings between them

Used to be able to quickly “understand” words



<u>Careful annotation of terms obviates need to learn meanings with data</u>

https://www.snomed.org/snomed-ct/five-step-briefing

Skipping natural language: Publish structured data

If you want to publish a paper about a crystal structure, you must publish it to PDB

Data must be in a machine-accessible format and available on the internet. No need for NLP!

```
    loop_
   _atom_site.group_PDB
   _atom_site.type_symbol
   _atom_site.label_atom_id
   _atom_site.label_comp_id
   _atom_site.label_asym_id
   _atom_site.label_seq_id
   _atom_site.label_alt_id
   _atom_site.cartn_x
   _atom_site.cartn_y
   _atom_site.cartn_z
   _atom_site.occupancy
   _atom_site.B_iso_or_equiv
   _atom_site.footnote_id
   _atom_site.auth_seq_id
   _atom_site.id
ATOM N  N   VAL  A  11 . 25.369  30.691  11.795  1.00  17.93 . 11 1
ATOM C  CA  VAL  A  11 . 25.970  31.965  12.332  1.00  17.75 . 11 2
ATOM C  C   VAL  A  11 . 25.569  32.010  13.881  1.00  17.83 . 11 3
#          [data omitted]
```

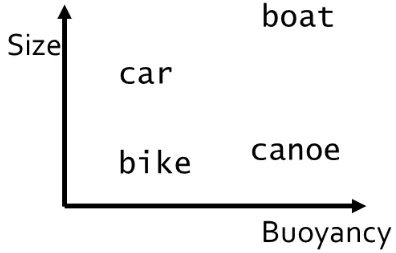
Figure: Bourne et al. Methods in Enzy. (1997)

Take away points

Core Concepts in NLP

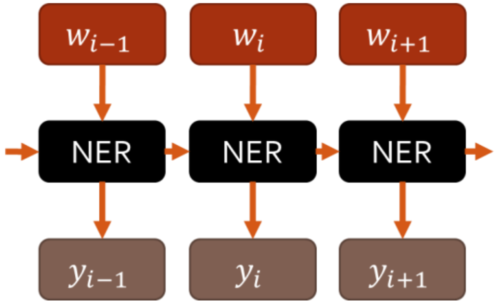
Many individual challenges

- Named-entity recognition
- Association mapping
- Translation
- Question Answering



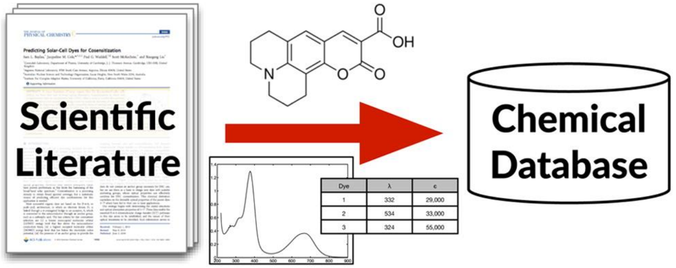
Two core methods:

1. Embeddings
2. Recurrent networks



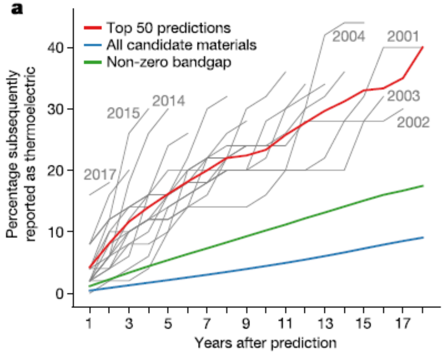
Core Applications

Information Extraction



Source: Cole and Swain. JCIM (2016)

Unsupervised Learning



Source: Tshitoyan et al. Nature (2019)