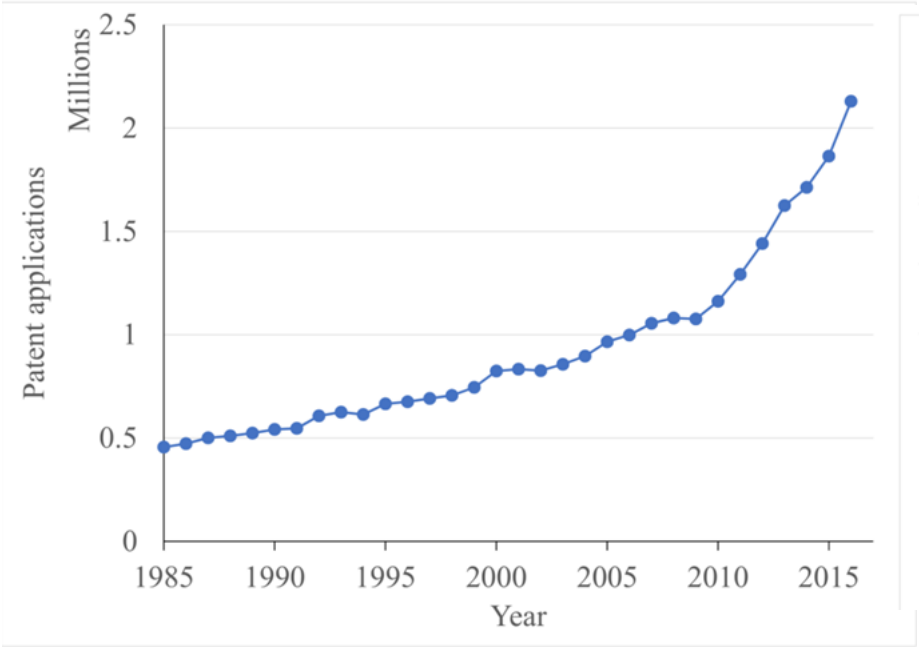
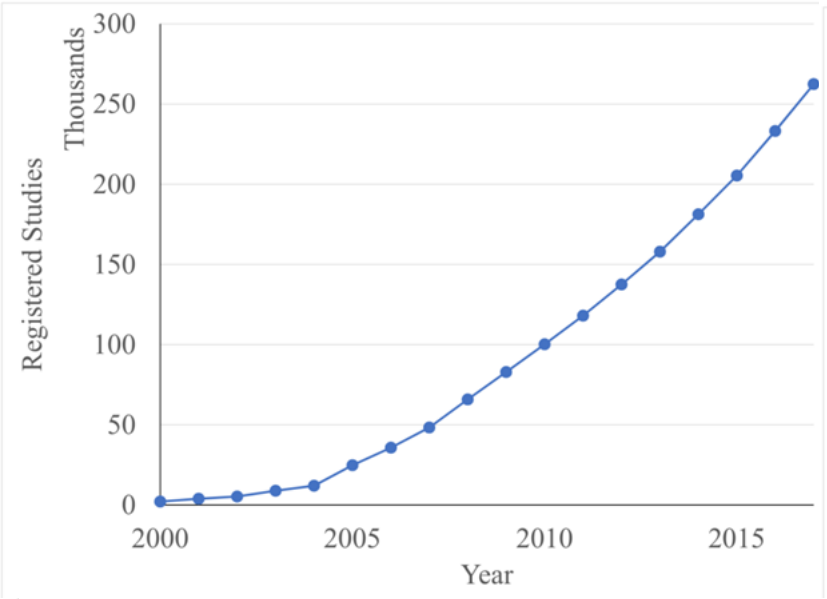
12 – Biomedical Text Mining

André Lamúrias

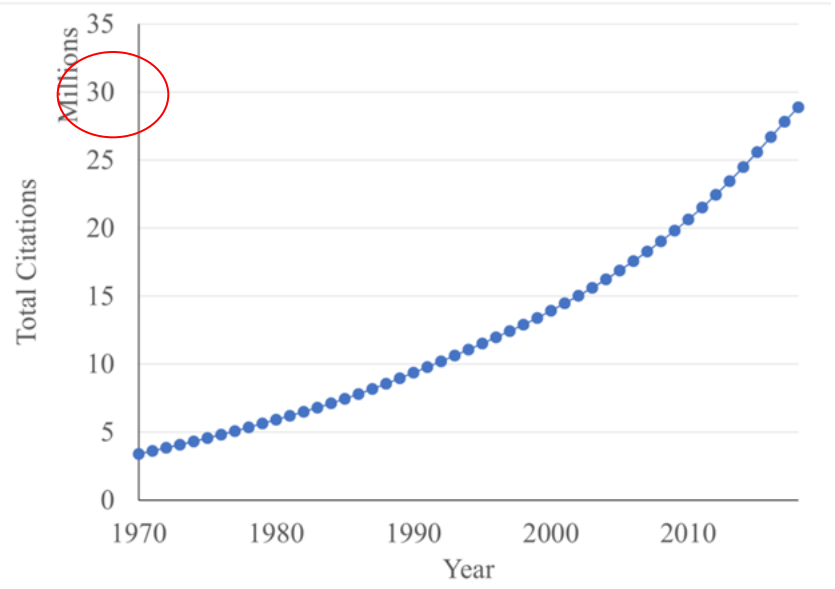
Biomedical Text Mining



WIPO

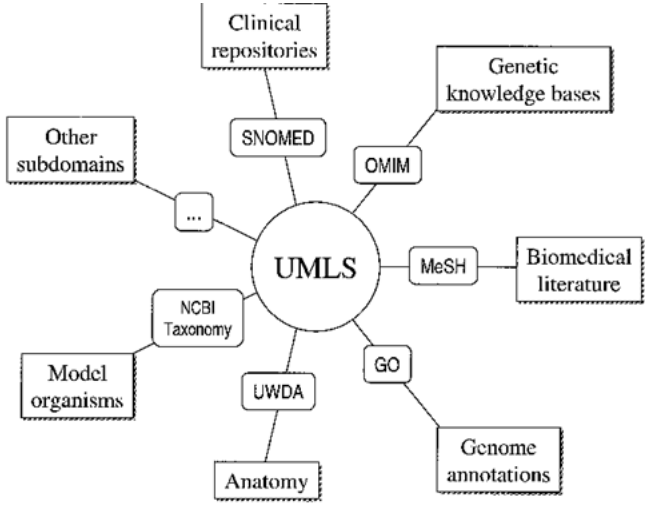


ClinicalTrials.Gov



MEDLINE/PubMed

DISEASE ONTOLOGY



GENEONTOLOGY
Unifying Biology

ChEBI

human phenotype ontology

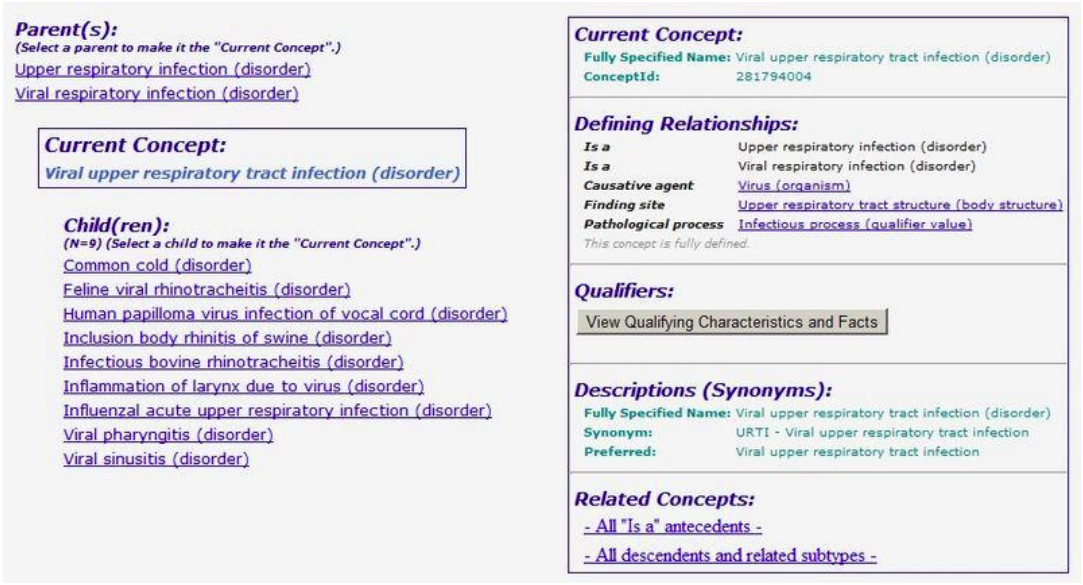
UniProt

KEGG Kyoto Encyclopedia of Genes and Genomes

Ensembl

FlyBase

RNAcentral



Parent(s):
(Select a parent to make it the "Current Concept".)
Upper respiratory infection (disorder)
Viral respiratory infection (disorder)

Current Concept:
Viral upper respiratory tract infection (disorder)

Child(ren):
(N=9) (Select a child to make it the "Current Concept".)
Common cold (disorder)
Feline viral rhinotracheitis (disorder)
Human papilloma virus infection of vocal cord (disorder)
Inclusion body rhinitis of swine (disorder)
Infectious bovine rhinotracheitis (disorder)
Inflammation of larynx due to virus (disorder)
Influenzal acute upper respiratory infection (disorder)
Viral pharyngitis (disorder)
Viral sinusitis (disorder)

Current Concept:
Fully Specified Name: Viral upper respiratory tract infection (disorder)
ConceptId: 281794004

Defining Relationships:

Is a	Upper respiratory infection (disorder)
Is a	Viral respiratory infection (disorder)
Causative agent	Virus (organism)
Finding site	Upper respiratory tract structure (body structure)
Pathological process	Infectious process (qualifier value)

This concept is fully defined.

Qualifiers:
View Qualifying Characteristics and Facts

Descriptions (Synonyms):
Fully Specified Name: Viral upper respiratory tract infection (disorder)
Synonym: URTI - Viral upper respiratory tract infection
Preferred: Viral upper respiratory tract infection

Related Concepts:
- All "Is a" antecedents -
- All descendents and related subtypes -

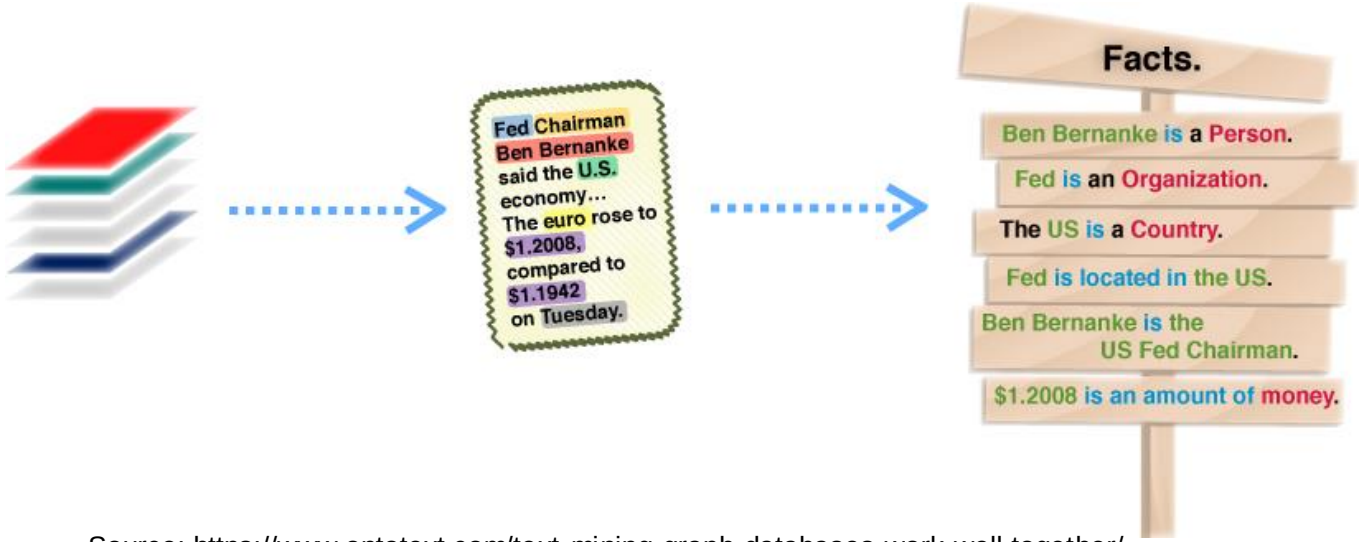
50 YEARS OMIM
Human Genetics Knowledge for the World

miRBase

Databases and ontologies

Text Mining

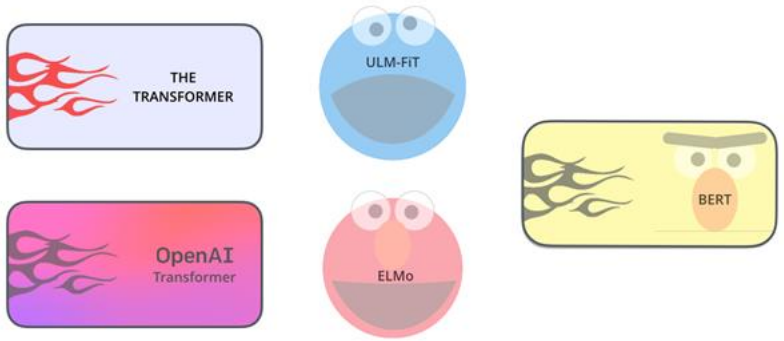
- Extract high quality information from text
- Evaluation using gold standards
- Techniques
  - Rule based
  - Machine learning
  - Deep Learning

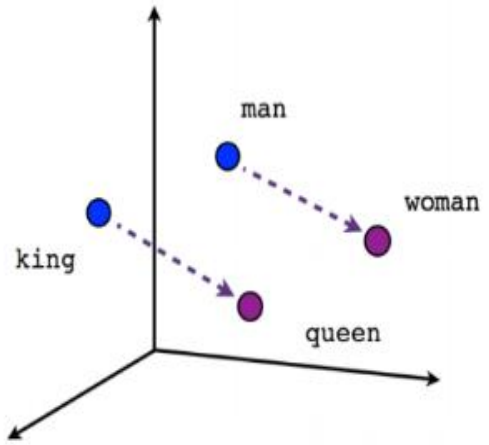


Source: https://www.ontotext.com/text-mining-graph-databases-work-well-together/

Recent developments in NLP

- 2013 – 2016: Word2vec/GloVe – word embeddings
- 2017 – 2019: Attention/Transformers/ELMo/BERT
- 2019 – ?: *BERT/GPT/LLMs





Male-Female

https://jalammar.github.io/illustrated-bert/ (2018)

Identifying Relevant information

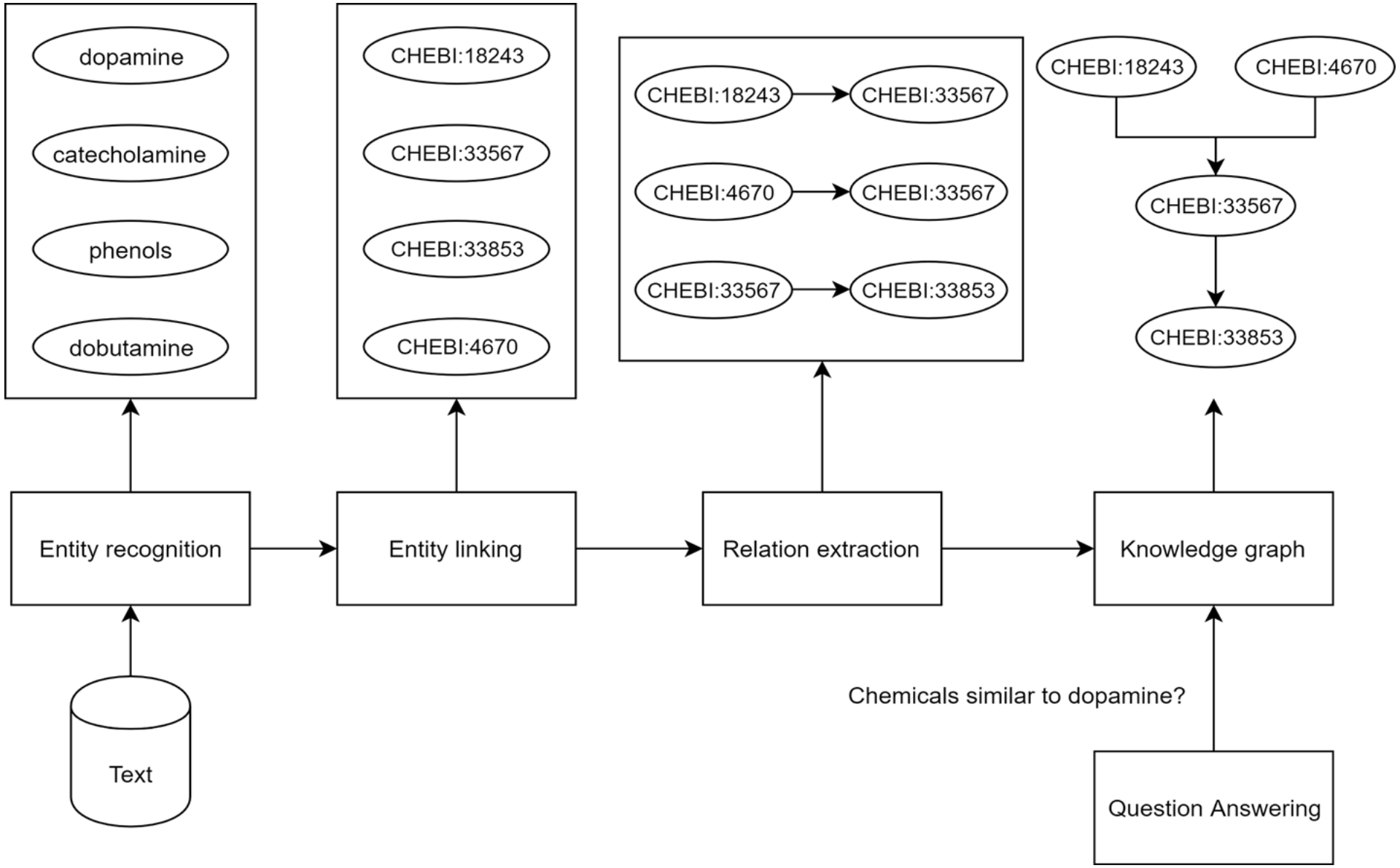
The CFTR gene displays a tightly regulated tissue-specific and temporal expression. Mutations in this gene cause cystic fibrosis (CF). In this study we wanted to identify trans-regulatory elements responsible for CFTR differential expression in fetal and adult lung and to determine the importance of inhibitory motifs in the CFTR-3'UTR with the aim of developing new tools for the correction of disease-causing mutations within CFTR. We show that lung development-specific transcription factors (FOXA, C/EBP) and microRNAs (miR-101, miR-145, miR-384) regulate the switch from strong fetal to very low CFTR expression after birth. By using miRNome profiling and gene reporter assays, we found that miR-101 and miR-145 are specifically upregulated in adult lung and that miR-101 directly acts on its cognate site in

CFTR - Gene ID:1080, P13569
miR-101 - Gene ID: 406893, miRBase MI0000103

(FOXA, C/EBP, miR-101,145,384) -> regulate -> CFTR

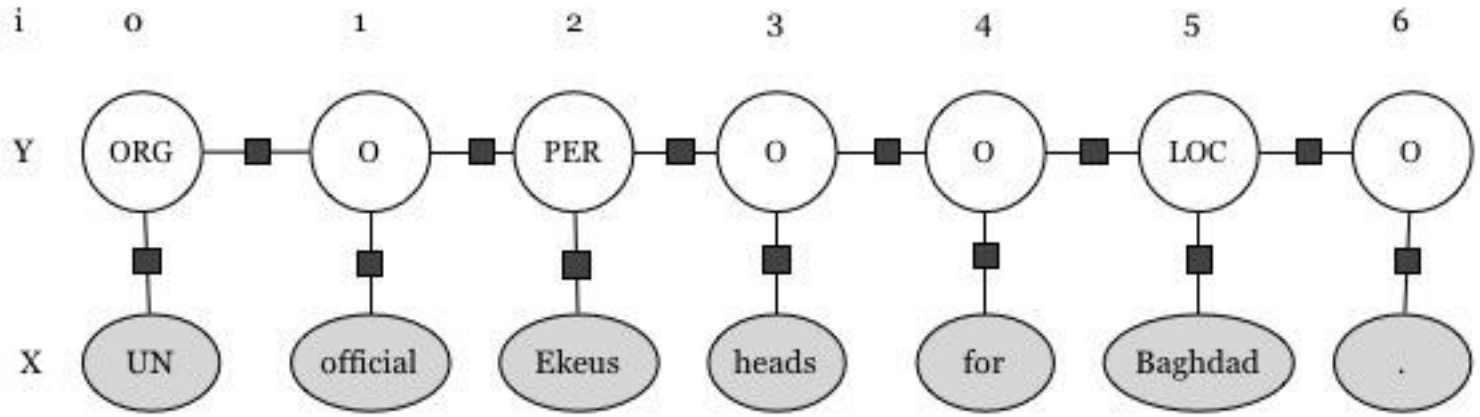
- Entities:
  - Genes, microRNAs, events, organs, disease, tools/techniques
- Database IDs
  - Entrez, UniProt, gene ontology, miRBase
- Relations
  - Between genes, miRNAs, TFs, etc

Pipeline example



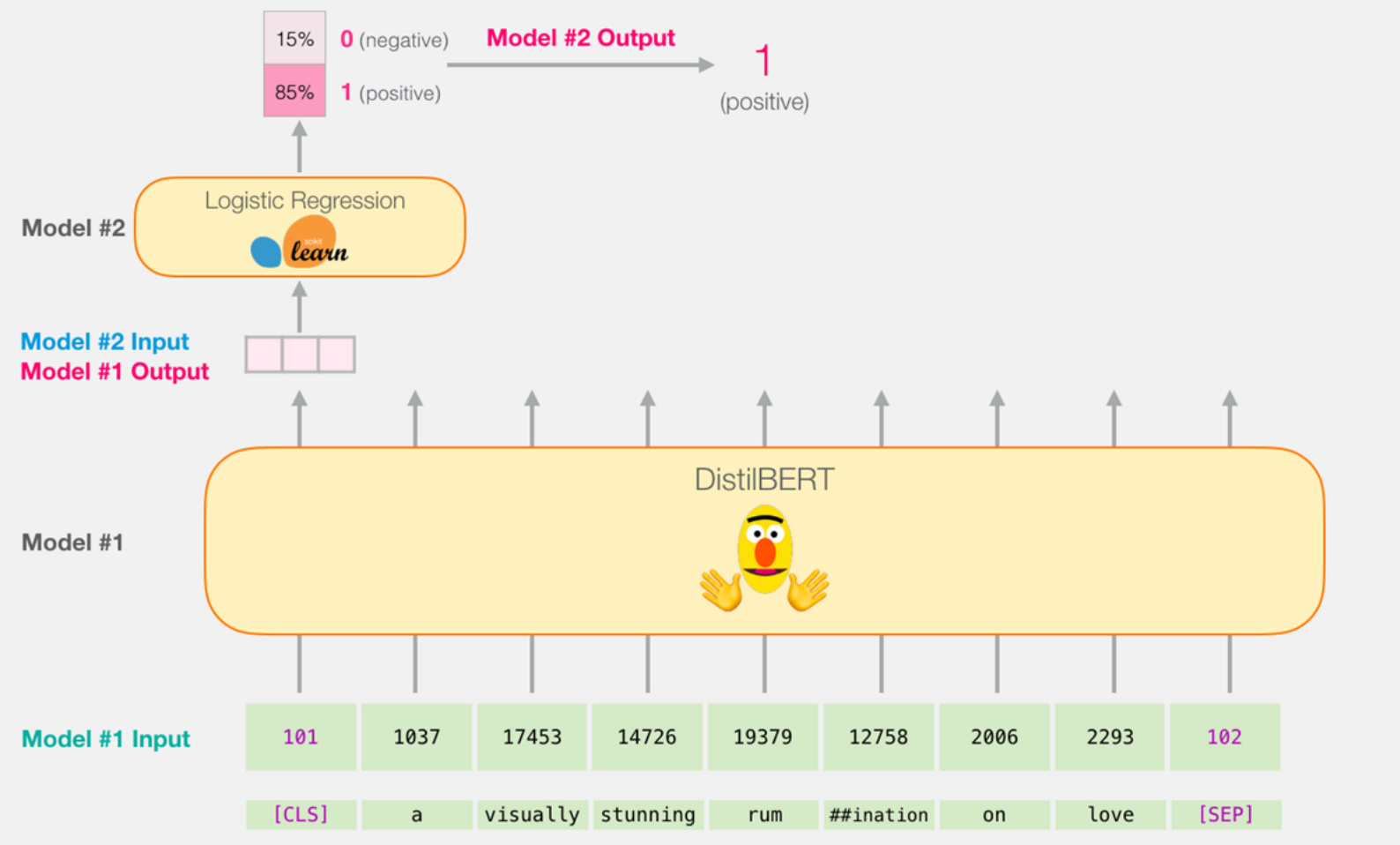
Named entity recognition (NER)

- Identify tokens that refer to entities of interest
- Entity types for biology: gene, chemical, disease, species
- Entities can be used to index documents, extract relations, link to external databases and ontologies
- No universal rules due to nomenclature variability and other factors



Tags	Description
B-PER	The beginning of a Person's name
I-PER	Part of a person's name
B-LOC	The beginning of a Location name
I-LOC	Part of a Location name
B-ORG	The beginning of a Organization name
I-ORG	Part of a Organization name
O	Not named-entity

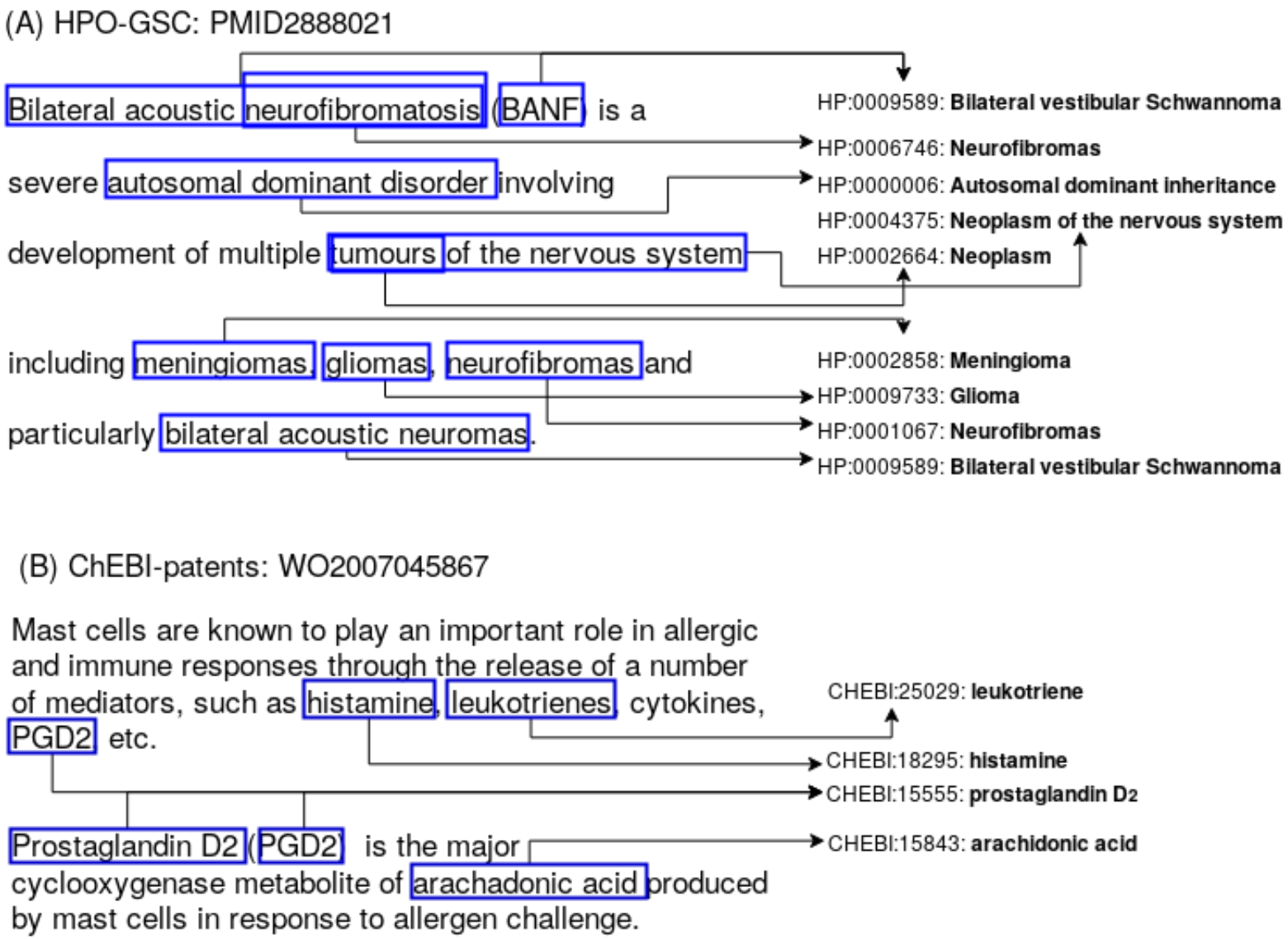
Deep learning/Transformer Models



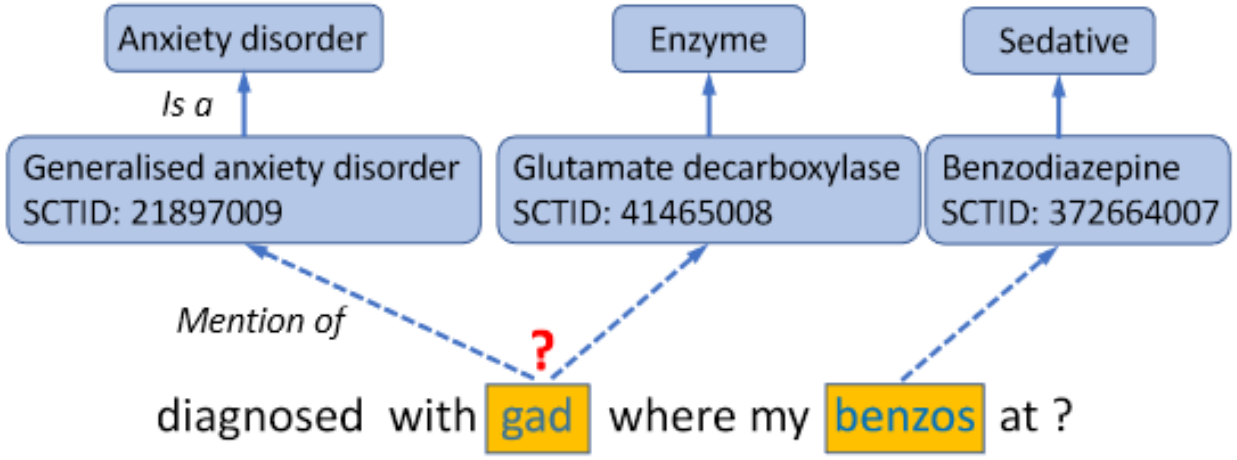
- Pretrained language model generates contextual word embeddings
- Classification head

http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Entity Linking (EL)



EL using deep learning



Basaldella, Marco, et al. "COMETA: A Corpus for Medical Entity Linking in the Social Media." *arXiv preprint arXiv:2010.03295* (2020).

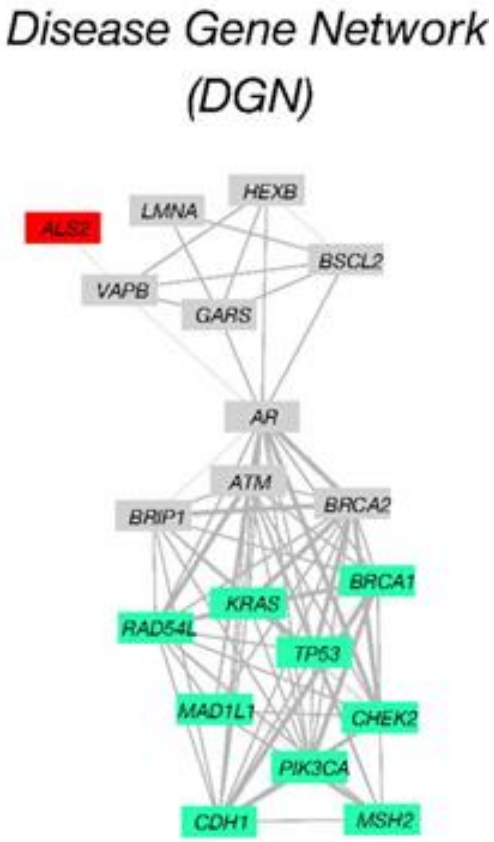
- Train vector representations of entity and concepts using BERT:
  - "gad" and "benzos" - use contextual representations
  - Concepts: use label and description
- Calculate score between each entity mention and each concept and choose concept with maximum score

score("gad", "generalised anxiety disorder SCTID: 21897009") = 0.8

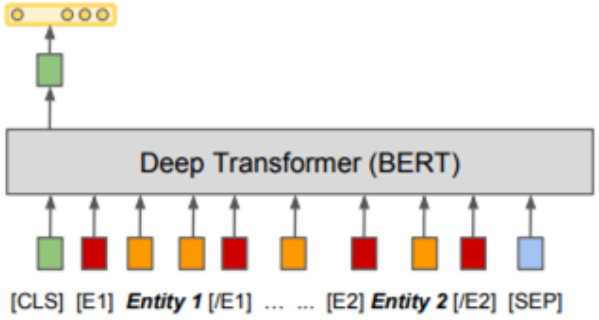
score("gad", "Glutamate decarboxylase SCTID: 41465008") = 0.1

Relation extraction (RE)

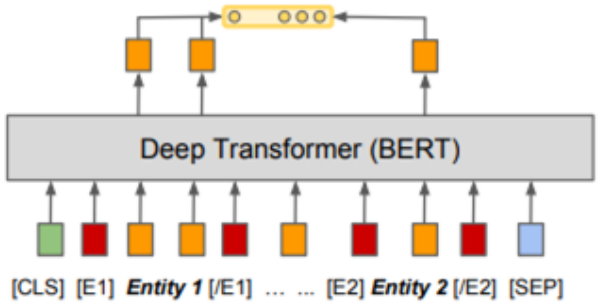
- For each two entities in a sentence, classify if a relation is described between them
- However the relation described may be between **more than 2 entities** and **across various sentences**
- More complex than NER, **more difficult to get data**
- Examples: between chemicals (advice, mechanism, effect), temporal, protein-protein,



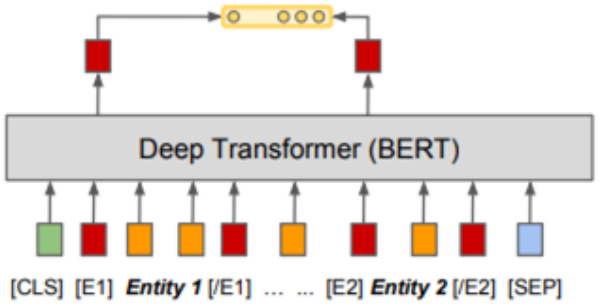
Relation Extraction



(d) ENTITY MARKERS – [CLS]



(e) ENTITY MARKERS – MENTION POOL.

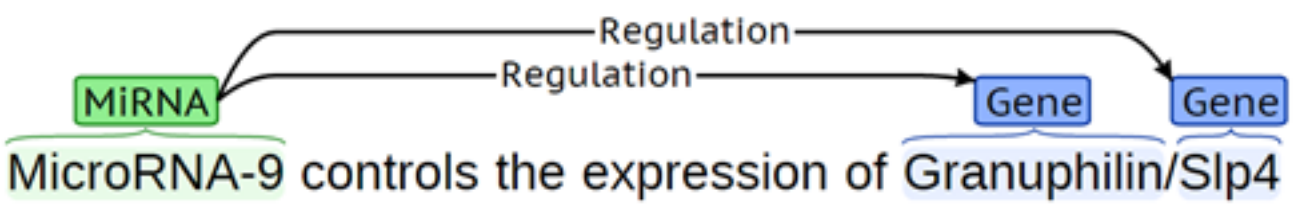


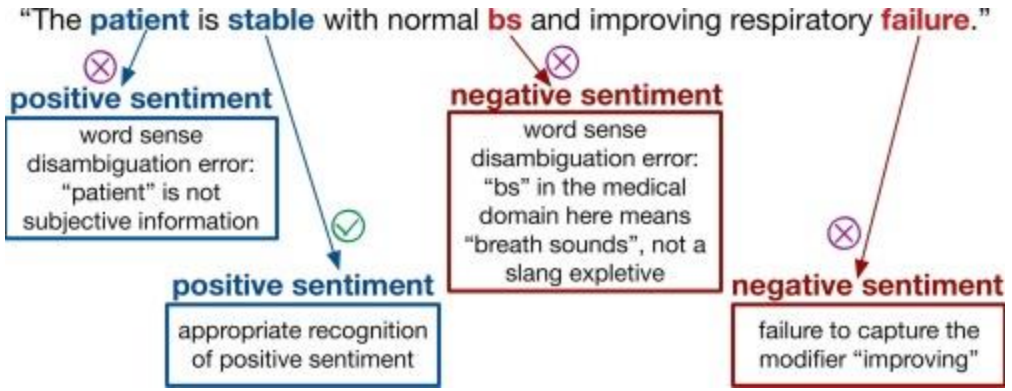
(f) ENTITY MARKERS – ENTITY START

Soares, Livio Baldini, et al. "Matching the Blanks: Distributional Similarity for Relation Learning." *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019.

Text mining tasks

- NER: Named Entity Recognition
- Entity Linking/Normalization
- RE: Relation Extraction
- ----------
- QA: Question answering
- Sentiment analysis
- Topic modeling
- Summarization





Weissman, Gary E., et al. "Construct validity of six sentiment analysis methods in the text of encounter notes of patients with critical illness." *Journal of biomedical informatics* 89 (2019): 114-121.

Shared tasks

- TREC: Text REtrievel Conference
  - Started in 1992
- SemEval: Semantic Evaluation
  - Started in 1998
- BioCreative
  - Critical Assessment of Information Extraction in Biology
  - Started in 2004; latest edition: 2023
- BioASQ
  - Started in 2013

TREC – Clinical Trials subtask

- Find clinical trials relevant for a patient
- Use synthetic patient cases (admission note)

```
<topics task="2022 TREC Clinical Trials">
  <topic number="-1">
    A 2-year-old boy is brought to the emergency department by his parents for 5 days of high fever
    and irritability. The physical exam reveals conjunctivitis, strawberry tongue, inflammation of
    the hands and feet, desquamation of the skin of the fingers and toes, and cervical
    lymphadenopathy with the smallest node at 1.5 cm. The abdominal exam demonstrates tenderness
    and enlarged liver. Laboratory tests report elevated alanine aminotransferase, white blood cell
    count of 17,580/mm, albumin 2.1 g/dL, C-reactive protein 4.5 mg, erythrocyte sedimentation rate
    60 mm/h, mild normochromic, normocytic anemia, and leukocytes in urine of 20/mL with no bacteria
    identified. The echocardiogram shows moderate dilation of the coronary arteries with possible
    coronary artery aneurysm.
  </topic>
</topics>
```

- Find relevant Clinical Trials from ClinicalTrials.gov (490,899 studies)
- https://www.trec-cds.org/2022.html

BioASQ

- Completed tasks:
  - Large-Scale Online Biomedical Semantic Indexing
  - MedProcNER On MEDical PROCedure Named Entity Recognition
  - DisTEMIST On Disease Text Mining And Indexing
  - MESINESP On Medical Semantic Indexing In Spanish
  - Funding Information Extraction From Biomedical Literature
- Ongoing (2024):
  - Biomedical Semantic QA (Involves IR, QA, Summarization And More)
  - Synergy On Biomedical Semantic QA For Developing Issues
  - MultiCardioNER On Mutiple Clinical Entity Detection In Multilingual Medical Content
  - BioNNE On Nested NER In Russian And English

Summary

- Biomedical Text mining
- Text mining tasks
  - Named Entity recognition
  - Entity Linking
  - Relation Extraction
- Text mining challenges
- Further reading
  - Speech and Language Processing Chapter 8
  - Text mining for bioinformatics using biomedical literature

Tutorial

- Use a BERT model with spacy to classify Named Entities and link to gene ontology
- Get embeddings of these entity and calculate their similarity
- https://spacy.io/

Tutorial

- Install transformers (use google colab with personal account: https://colab.research.google.com/drive)

```
!pip install scispacy
```

- Download a language model e.g. SciBERT:

```
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.4/en_core_sci_scibert-0.5.4.tar.gz
```

  - Check https://allenai.github.io/scispacy/ for other models if you're using your computer

Tutorial

```
import spacy
import scispacy
import scispacy.linking
nlp = spacy.load("en_core_sci_scibert")
doc = nlp("""Alterations in the hypocretin
receptor 2 and preprohypocretin genes produce
narcolepsy in some animals.""")
```

```python
# from https://applied-language-technology.mooc.fi/html/notebooks/part_iii/05_embeddings_continued.html
from spacy.language import Language
import numpy as np

@Language.factory('tensor2attr')
class Tensor2Attr:
    def __init__(self, name, nlp):
        pass

    def __call__(self, doc):
        self.add_attributes(doc)
        return doc

    def add_attributes(self, doc):

        doc.user_hooks['vector'] = self.doc_tensor
        doc.user_span_hooks['vector'] = self.span_tensor
        doc.user_token_hooks['vector'] = self.token_tensor
        doc.user_hooks['similarity'] = self.get_similarity
        doc.user_span_hooks['similarity'] = self.get_similarity
        doc.user_token_hooks['similarity'] = self.get_similarity

    def doc_tensor(self, doc):
        return doc._.trf_data.tensors[-1].mean(axis=0)

    def span_tensor(self, span):
        tensor_ix = span.doc._.trf_data.align[span.start: span.end].data.flatten()
        out_dim = span.doc._.trf_data.tensors[0].shape[-1]
        tensor = span.doc._.trf_data.tensors[0].reshape(-1, out_dim)[tensor_ix]
        return tensor.mean(axis=0)

    def token_tensor(self, token):
        tensor_ix = token.doc._.trf_data.align[token.i].data.flatten()
        out_dim = token.doc._.trf_data.tensors[0].shape[-1]
        tensor = token.doc._.trf_data.tensors[0].reshape(-1, out_dim)[tensor_ix]
        return tensor.mean(axis=0)

    def get_similarity(self, doc1, doc2):
        return np.dot(doc1.vector, doc2.vector) / (doc1.vector_norm * doc2.vector_norm)
```

```
nlp.add_pipe('tensor2attr')

doc = nlp("""Alterations in the hypocretin receptor 2 and
preprohypocretin genes produce narcolepsy in some animals.""")

for token in doc:

    print(token.text, token.lemma_, token.pos_, token.tag_,
token.dep, token.shape_, token.is_alpha, token.is_stop,
token.vector[:5])

print(doc[0].vector.shape)
```

More efficient:

```
texts = ["Alterations in the hypocretin receptor 2 and preprohypocretin genes produce narcolepsy in some animals.",
         "Glaucoma is a leading cause of blindness but its molecular etiology is poorly understood.",
         "Glaucoma involves retinal ganglion cell death and optic nerve damage that is often associated with elevated intraocular pressure (IOP)"]
for doc in nlp.pipe(texts, disable=["tok2vec", "tagger", "parser", "attribute_ruler", "lemmatizer"]):
    # Do something with the doc here
    print([(ent.text, ent.label_, ent.vector[:3]) for ent in doc.ents])
```

Download a document

```
!curl -s "https://raw.githubusercontent.com/UCDenver-ccp/CRAFT/master/articles/txt/11532192.txt" > doc.txt
with open("doc.txt") as f:
  doc_text = f.read()
print(doc_text)
entities = []
for doc in nlp.pipe(doc_text.split("\n"), disable=["tok2vec", "tagger", "parser", "attribute_ruler", "lemmatizer"]):
  print(doc.ents)
```

Entity Linking using MER

```
!apt-get install gawk
!pip install merpy ssmpy
import merpy
merpy.download_lexicons()
```

Get linked entities of each text

```
def get_doc_entities(doc):
  entities = [] # store tuples (name, ID, vector)
  entity = doc.ents
  for ent in entity:
    linked_ent = merpy.get_entities(ent.text, "go")
    #print(ent, ent[0].ent_type_, linked_ent)
    if len(linked_ent[0]) > 1:
      print(ent, linked_ent)
      entities.append((linked_ent[-1][-2], linked_ent[-1][-1].split("/")[-1], ent.vector))
  return entities
```

Get linked entities of each text

```
entities = []
for doc in nlp.pipe(doc_text.split("\n"),
disable=["tok2vec", "tagger", "parser",
"attribute_ruler", "lemmatizer"]):

    sent_entities = get_doc_entities(doc)
    if len(sent_entities) > 0:
        entities += sent_entities
```

Compare entities from the same document

```
from sklearn.metrics.pairwise import cosine_similarity

# compute embedding sim between every pair
for ent1 in entities:
  for ent2 in entities:
    print(ent1[0], "x", ent2[0], "=", cosine_similarity([ent1[2]], [ent2[2]]))
```

Better:

```
all_sims = cosine_similarity([e[2] for e in entities],[e[2] for e in entities])
for i, ent1 in enumerate(entities):
  for j, ent2 in enumerate(entities):
    print(ent1[0], "x", ent2[0], "=", all_sims[i][j])
```

Combine entities of the same GO term

```
go2emb = {}
for ent in entities:
  if ent[1] not in go2emb:
    go2emb[ent[1]] = []
  go2emb[ent[1]].append(ent[2])

for goid in go2emb:
  go2emb[goid] = sum(go2emb[goid])/len(go2emb[goid])
for goid in go2emb:
  for goid2 in go2emb:
    print(goid, "x", goid2, "=",
          cosine_similarity([go2emb[goid]], [go2emb[goid2]]))
```

Tutorial

- Now do this for more documents:
  - Get articles from PubMed or use these files:
    - https://raw.githubusercontent.com/UCDenver-ccp/CRAFT/master/articles/txt/11319941.txt
    - https://raw.githubusercontent.com/UCDenver-ccp/CRAFT/master/articles/txt/11597317.txt
  - What are the most similar entity pairs of each document? What about against other documents?
  - What is the average entity similarity?
  - What are the most common GO terms in a document and in all documents?

Assignment 2

- Check CRAFT corpus test set PMIDs:
  - https://github.com/UCDenver-ccp/CRAFT/blob/master/articles/ids/craft-ids-test.txt
- Select 5 to 10 documents
- Apply NER and EL pipeline from this tutorial
- Calculate semantic similarity using ontology and using embeddings
  - Between every term of the same document and between every document
- Analyze results – find compare scores and find outliers