CS 533: Natural Language Processing

Knowledge-Intensive Language Tasks

Karl Stratos

Rutgers University

Semi-Supervised Learning with VAEs

- Limited labeled data $L = \{(x_1, y_1) \ldots (x_N, y_N)\}$
- Unlabeled data $U = \{x'_1 \ldots x'_M\}$ where $M \gg N$
- LVGM (Kingma et al., 2014)

$$p_\theta(y, z, x) = \pi_\theta(y) \times \mathcal{N}(0_d, I_{d \times d})(z) \times \kappa_\theta(x|y, z)$$

  Label y treated as latent on U ($z \in \mathbb{R}^d$ always latent)

- Inference network

$$q_\phi(y, z|x) = q_\phi(y|x) \times \mathcal{N}(\mu_\phi(x, y), \mathrm{diag}(\sigma^2_\phi(x)))(z)$$

- "ELBO-regularized classification": Maximize

$$\widehat{\mathrm{ELBO}}^{L}_{XY}(\theta, \phi) + \widehat{\mathrm{ELBO}}^{U}_{X}(\theta, \phi) + \frac{\alpha}{N} \sum_{i=1}^{N} \log q_\phi(y_i|x_i)$$

  Use $q_\phi(y|x)$ as classifier, $\kappa_\theta(x|y, z)$ as label-conditional generator

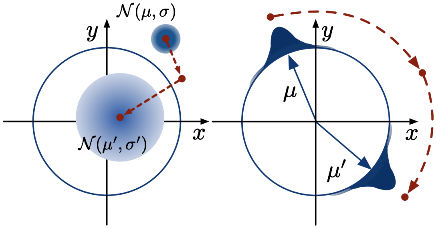
Prior-Based Approach to Posterior Collapse

- Recall: Can kill KL by setting $q_\phi(z|x) = \pi(z)$

$$\mathrm{ELBO}(\theta, \phi) = \mathop{\mathbf{E}}_{x\sim\mathbf{pop},\ z\sim q_\phi(\cdot|x)}\left[\log \kappa_\theta(x|z)\right] - \cancel{\mathop{\mathbf{E}}_{x\sim\mathbf{pop}}\left[D_{\mathrm{KL}}(q_\phi(\cdot|x)||\pi)\right]}$$

  - Undesirable local optimum: Decoder κ_θ will ignore z
  - Many tricks to enforce large KL (annealing, free bits)

- Another solution: Use a prior that's difficult to zero in. Example: Uniform distribution over (hyper)sphere



(Xu and Durrett, 2018)

Impossible to match with any fixed finite variance

von Mises-Fisher (vMF) VAE

- Sphere in $\mathbb{R}^d$: $S^{d-1} = \left\{z \in \mathbb{R}^d : ||z|| = 1\right\}$
- vMF with mean $\mu \in S^{d-1}$ and “concentration” $\kappa \geq 0$

$$\mathrm{vMF}(\mu, \kappa)(z) = \frac{\kappa^{\frac{2}{d}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)} \exp(\kappa \mu^\top z) \qquad \forall z \in S^{d-1}$$

  Uniform if $\kappa = 0$ regardless of μ (e.g., $\frac{1}{2\pi}$ for $d = 2$)

  - Set $\pi = \mathrm{vMF}(\cdot, 0)$
  - Set $q_\phi(\cdot|x) = \mathrm{vMF}(\frac{\mu_\phi(x)}{||\mu_\phi(x)||}, \kappa)$ for some fixed $\kappa > 0$
  - Then $D_{\mathrm{KL}}(q_\phi(\cdot|x)||\pi) = C(\kappa)$ for some positive constant $C(\kappa)$

- Also have a (complicated) reparameterization trick (Wood, 1994)

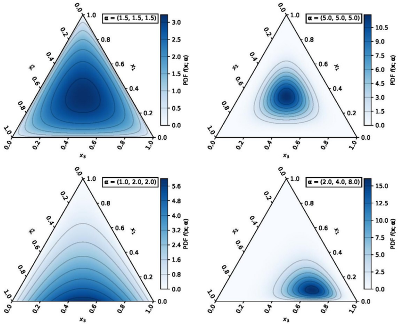
$$z \sim \mathrm{vMF}\left(\frac{\mu_\phi(x)}{||\mu_\phi(x)||}, \kappa\right) \quad \Leftrightarrow \quad z = \epsilon_\kappa \frac{\mu_\phi(x)}{||\mu_\phi(x)||} + \sqrt{1 - \epsilon_\kappa^2} v_\kappa$$

  where $v_\kappa \perp \mu_\phi(x)$ and ϵ_κ are appropriate random noise

- Less susceptible to posetrior collapse, better likelihood/representation (Xu and Durrett, 2018)

Enforcing Prior Sparsity

- Assume categorical latent $z \in \{1 \ldots K\}$, want to make prior over z sparse (i.e., support size $\ll K$)
- Example: Non-parametric language model that retrieves a training sentence z and edits (Guu et al., 2018), expensive to consider all data
- Solution: Have a sparsity-encouraging prior over the prior



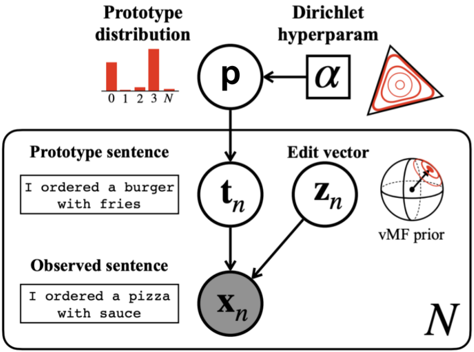
(Image credit: Wikipedia)

- **Dirichlet distribution**: $\mathrm{Dir}_\alpha(p) \propto \prod_{z=1}^{K} p(z)^{\alpha_z}$
- A density over categorical distributions, sparsity controllable by $\alpha \in \mathbb{R}^K$
- $\mathrm{Dir}_{<1_K}$ concentrated at point-mass distributions
- $\mathrm{Dir}_{>1_K}$ concentrated at uniform distributions

Example: Sparse Prototype Language Model (He et al., 2020)

- Observation: Sentence x
- Latents: "Prototype" distribution p over N training examples, $t \in \{1 \dots N\}$, edit vector $z \in \mathbb{R}^d$
- LVGM

$$p_{\alpha,\theta}(p, t, z, x) = \mathrm{Dir}_{\alpha}(p) \times p(t) \times \mathrm{vMF}(\cdot, 0)(z) \times \kappa_{\bar{\theta}}(x|z, t)$$



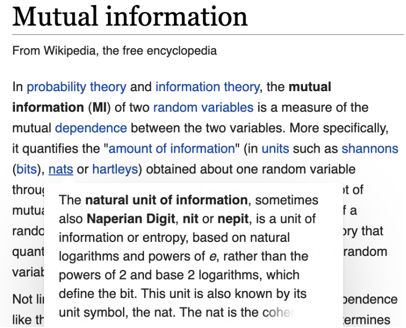
Wikipedia

- **Knowledge base (KB)**: Dataset storing complex information
- Quintessential KB: **Wikipedia**
  - Crowdsourced, constantly growing, free
  - Consists of **pages** organized into **namespaces** (Main, User, Wikipedia, Category, etc.)
  - Main namespace contains **articles** (aka. **entities**), comprehensive summaries of notable topics
    - Not all main pages are articles: Must exclude redirect (e.g., `UK` $\mapsto$ `United Kingdom`), disambiguation (e.g., `Mercury` lists `Mercury (planet)`, `Mercury (element)`, ...), front
- Multilingual: > 270 languages. Number of articles as of April 2021 (in millions)
  - English 6.2, German 2.6, French 2.3, Russian 1.7 ...
- Articles also labeled by (possibly multiple) **categories** organized as a directed graph, e.g., *Fields of mathematics*
  - Main article: `Areas of mathematics`
  - Parents: *Mathematics*, *Subfields by academic discipline*, ...
  - Children: *Algebra*, *Probability and statistics*, ...

Hyperlinks and Tables in Wikipedia

Wikilinks: Internal links mapping text spans to corresponding articles

Infobox: A table summarizing the article



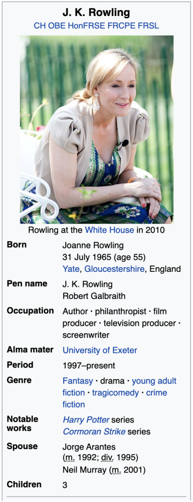
Mutual information

From Wikipedia, the free encyclopedia

In probability theory and information theory, the **mutual information** (**MI**) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" (in units such as shannons (bits), nats or hartleys) obtained about one random variable throu … nt of mutu … f a rando … ory that quant … random variab …

Not li … pendence like th … termines

The **natural unit of information**, sometimes also **Naperian Digit**, **nit** or **nepit**, is a unit of information or entropy, based on natural logarithms and powers of *e*, rather than the powers of 2 and base 2 logarithms, which define the bit. This unit is also known by its unit symbol, the nat. The nat is the cohe



J. K. Rowling

CH OBE HonFRSE FRCPE FRSL

Rowling at the White House in 2010

Born	Joanne Rowling 31 July 1965 (age 55) Yate, Gloucestershire, England
Pen name	J. K. Rowling Robert Galbraith
Occupation	Author · philanthropist · film producer · television producer · screenwriter
Alma mater	University of Exeter
Period	1997–present
Genre	Fantasy · drama · young adult fiction · tragicomedy · crime fiction
Notable works	*Harry Potter* series *Cormoran Strike* series
Spouse	Jorge Arantes (m. 1992; div. 1995) Neil Murray (m. 2001)
Children	3

Wikipedia in NLP

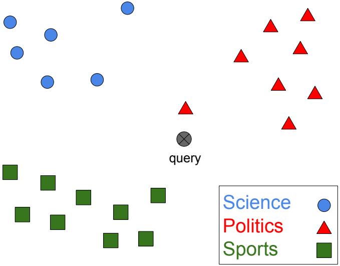
- High-quality corpus for unsupervised pretraining
  - English Wikipedia: 3.9 billion words (600 words/article)
  - Vast knowledge in multiple languages in standard form
  - Used by virtually all self-supervised representation learning methods in NLP (word embeddings, language models)
- Large-scale annotated dataset for entity linking (EL)
  - Map a span in text to the underlying entity
- Passages for question answering (QA) and fact checking
  - The system must support its prediction by using these passages
- More generally, we call tasks **knowledge-intensive** if they rely on a KB
  - Tasks generally not considered knowledge-intensive: Topic classification, sentiment analysis, translation
- Other KBs: Wikias (encyclopedias on fictional works like *Star Wars*), PubMed (32 million citations for biomedical literature), Wikidata (relation instances (e_1, r, e_2) tightly integrated with Wikipedia), private KBs (industry)

Text Retrieval

- When dealing with a KB, often need a *very* scalable method for retrieving entries of interest because there are too many
- EL: Given a document x and a mention span (i, j)
    1. Retrieve top-K "most relevant" Wikipedia entities $e_1 \ldots e_K$
    2. Predict $\arg\max_k \textbf{score}_\theta(x, i, j, e_k)$
- QA: Given a question q
    1. Retrieve top-K "most relevant" passages $p_1 \ldots p_K$ (e.g., from all blocks of 512 tokens in Wikipedia)
    2. Predict $\arg\max_{(i,j)} \textbf{score}_\theta(q, p_k, i, j)$ as answer string
- Retrieval system choices
    - Classical information retrieval (IR): TFIDF, BM25, PageRank. Effective and task-agnostic but can't improve by learning
    - Neural: Learn a parametric model for task-specific retrieval, must be extremely efficient

Document Representation

- Want "similar" documents closer to each other than "unsimilar" ones under some notion of distance/similarity



- Starting point: Naive bag-of-words (BOW) embedding

$$\text{📄} \quad \rightarrow \quad (0,0,0,1,\ldots,0,1,0,\ldots,0,0) \in \{0,1\}^V$$

  - Distance: Number of dimensions that differ ("Hamming")
  - Limitation: All term types weighted equally (e.g. "the" and "Microsoft" have equal weights)

TFIDF (Term Frequency, Inverse Document Frequency)

- Idea: A term in document matters less if it appears all the time in other documents
- Each document $d \in D$ represented as sparse $x(d, D) \in \mathbb{R}^V$

$$x_t(d, D) = \underbrace{[[t \in d]]}_{\substack{\mathrm{tf}(t,d) \\ \text{(can also be counts)}}} \times \underbrace{\log \frac{|D|}{|\{d' \in D :\ t \in d'\}|}}_{\mathrm{idf}_D(t)}$$

- Note the dot product

$$x(d, D)^\top x(d', D) = \sum_{t \in \mathcal{V}} \mathrm{tf}(t, d) \times \mathrm{tf}(t, d') \times \mathrm{idf}_D(t)^2$$

- Use cosine similarity and cosine distance (bounded in $[0, 1]$ because all terms nonnegative)

$$\cos_D(d, d') = \frac{x(d, D)^\top x(d', D)}{||x(d, D)||\ ||x(d', D)||} \quad \mathrm{dist}_D(d, d') = 1 - \cos_D(d, d')$$

BM25 (Best Match 25)

- Idea: TFIDF with smoothing + document length modeling
- BM25 score between a query q and a candidate document $d \in D$

$$\mathrm{BM25}_{D,\alpha,\beta}(d,q) = \sum_{t \in q} \mathrm{tf}^{\mathrm{BM25}}_{\alpha,\beta}(t,d) \times \mathrm{idf}^{\mathrm{BM25}}_{D}(t)$$

where for some α, β and average document length L^{D}_{avg}

$$\mathrm{tf}^{\mathrm{BM25}}_{\alpha,\beta}(t,d) = \frac{\textbf{count}(t,d)(\alpha+1)}{\textbf{count}(t,d) + \alpha(1 - \beta + \beta(\left|d\right| / L^{D}_{\mathrm{avg}}))}$$

$$\mathrm{idf}^{\mathrm{BM25}}_{D}(t) = \log \frac{\left|D\right| - \left|\{d' \in D :\ t \in d'\}\right| + 0.5}{\left|\{d' \in D :\ t \in d'\}\right| + 0.5}$$

- Currently the go-to choice for IR
- BOW, TFIDF, BM25: Can be generalized to n-gram vectors

Dual Encoder

- Simplest form of parametric retriever, defines similarity between two texts x, y by

$$\mathrm{Dual}_\theta(x,y) = \underbrace{\textbf{enc}_\theta^{(1)}(x)}_{\text{dense vector in } \mathbb{R}^d} \cdot \underbrace{\textbf{enc}_\theta^{(2)}(y)}_{\text{dense vector in } \mathbb{R}^d}$$

- If $\textbf{enc}_\theta^{(1)} = \textbf{enc}_\theta^{(2)}$ called "siamese" network
- Crucially, similarity search with dense vectors can be done very efficiently (more on this later)
  - Implication: Can precompute embeddings at test time for efficient inference
  - Still not efficient enough for training however
- Common parameterization by pretrained LMs, e.g., [CLS] embeddings of two independent BERTs

$$\textbf{enc}_\theta^{(1)}(x) = \mathrm{BERT}_\theta^{(1)}(\text{[CLS]}\ x)[0] \in \mathbb{R}^d$$
$$\textbf{enc}_\theta^{(2)}(y) = \mathrm{BERT}_\theta^{(2)}(\text{[CLS]}\ y)[0] \in \mathbb{R}^d$$

Training by Noise Contrastive Estimation (NCE)

- Training data: $(x_1, y_1) \ldots (x_N, y_N)$ where x is query text and y is target KB entry (itself text, e.g., entity description, passage). The space of y is huge (e.g., all possible texts of length 512), so can't just optimize softmax-loss
- NCE: An approximation to softmax-loss

$$J_{\mathsf{NCE}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\mathrm{Dual}_\theta(x_i, y_i))}{\sum_{k=1}^{K} \exp(\mathrm{Dual}_\theta(x_i, y_{i,k}))}$$

  where $y_{i,1} \ldots y_{i,K} \sim \mathrm{noise}$ are negative examples for (x_i, y_i) drawn from some "noise" distribution
- With suitable K (hyperparameter), training efficient independently of how large the space of y is
- NCE only for training: At test time consider all KB entries
- Lots of theoretical results on the properties of J_{NCE}, depend on the choice of noise and K

Approximate Neareast Neighbor Search

- "Keys" $\mathcal{Y} \subset \mathbb{R}^d$, huge $M := |\mathcal{Y}|$ (possibly billions)
- Goal: Given query $x \in \mathbb{R}^d$ and number of neighbors $K \leq M$ compute

  $$\mathcal{Y}(x) \in K\text{-}\underset{y \in \mathcal{Y}}{\text{argmin}}\ ||x - y||$$

  Boils down to matrix multiplication since $||x - y||^2 = ||x||^2 + ||y||^2 - 2x^\top y$
- Exact search: Sort $||x - y||$ for all $y \in \mathcal{Y}$, return top-K. Time/memory complexity linear in M
- Approximate search based on **quantization**

  $$y \approx q_1(y) + q_2(y - q_1(y))$$

- q_1, q_2 "quantize" (i.e., cluster) $\mathbb{R}^d$ into finite sets
- Idea: Use q_1 to reduce search space, use q_2 to make up for error
- Efficient multi-GPU implementations available, e.g., Faiss (Johnson et al., 2017)

Two-Stage Quantization

- **Precompute.**
  - Cluster $\mathcal{Y}$ into $\mathcal{I}_1 \ldots \mathcal{I}_{\sqrt{M}}$. Set $q_1(y) = \text{centroid}(\mathcal{I}_{i(y)})$
  - Cluster subvectors $y = (y^1 \ldots y^b)$ into 256 groups for some $b = \{4, 8, \ldots, 64\}$. Each y is assigned one of 256^b possible clusters indexed by $\sum_{n=0}^{b-1} 256^n q^n(y^n)$. Set $q_2(y)$ to be the centroid of that cluster.
- **Test time.** Given query $x \in \mathbb{R}^d$ and K,
  - Find τ most promising clusters to focus on

$$C_\tau(x) \in \underset{i=1 \ldots \sqrt{M}}{\tau\text{-argmin}} \; ||x - \text{centroid}(\mathcal{I}_i)||$$

  - Search over these clusters for K nearest neighbors using quantized values

$$\widehat{\mathcal{Y}}(q) \in \underset{y \in \cup_{i \in C_\tau(x)} \mathcal{I}_i}{K\text{-argmin}} \; ||x - q(y)||$$

    where the distance is computed efficiently by working with decomposed form $y = (y^1 \ldots y^b)$

Retrieve-and-Rerank Approach

- Often effective to rerank candidates retrieved by a fast retriever with a more powerful model
- EL (Wu et al., 2020): Link mention m (with left/right context) to Wikipedia entity e (title + first paragraph)
  1. Train a dual encoder retriever by NCE

  $$\mathrm{Dual}_\theta(m, e) = \mathrm{BERT}_\theta^{(1)}(\text{[CLS]}\ m)[0] \cdot \mathrm{BERT}_\theta^{(2)}(\text{[CLS]}\ e)[0]$$

  2. Use the retriever to get top-K (e.g., 64) highest-scoring entities $e^{(1)} \ldots e^{(K)}$. Define a *joint* model by

  $$\mathrm{Joint}_\theta(m, e) = \underbrace{w^\top}_{\text{projection to scalars}} \mathrm{BERT}_\theta^{(3)}(\underbrace{\text{[CLS]}\ m\ \text{[SEP]}\ e}_{\text{deep cross-attention}})[0]$$

  Can be trained by optmizing $\log \mathrm{softmax}_1((m, e^{(k)})_{k=1}^K)$ where we set $e^{(1)}$ to be gold
- Significant improvement if the task requires some reading comprehension (i.e., easy to get high top-K recall but hard to get high accuracy). Can also be trained jointly (Lee et al., 2019)

Span-Selection Model

- Often need to predict span (i, j) in passage p given question q (many tasks can be reduced to QA)

$q =$When was Rutgers founded?

$p =$Rutgers University is a public land-grant research university based in New Brunswick,

New Jersey. Chartered on **November 10, 1766** Rutgers was originally . . .

$(i, j) =(17, 19)$

- Convention: Prepend a special symbol and set $i = j = 0$ if no answer can be found
- Simple span-selection model by conditional independence assumption

$$p_\theta(i, j|p, q) = s_\theta(i|p, q) \times t_\theta(j|p, q)$$

Reasonble given enough transformation on p, q to render i, j conditionally independent

Example: BERT-based NQ Model (Alberti et al., 2019)

- (Simplified) NQ Input: Question q, passages $p_1 \ldots p_n$ of a Wikipedia article (assumed to contain an answer somewhere)
- For each (p_k, q) define a distribution over $(0, 1, \ldots |p_k|)$ by

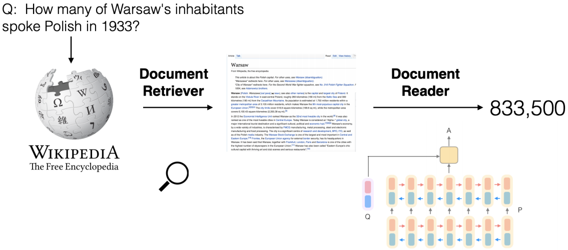
$$s_\theta(\cdot|p,q) = \mathrm{softmax}(\underbrace{w_s^\top}_{\text{projection to scalars}} \textbf{BERT}(\underbrace{\text{[CLS] } p_k \text{ [SEP] } q}_{\text{deep cross-attention}}) \underbrace{[: |p_k| + 1]}_{\text{normalize over passage positions}})$$

  Similarly for $q_\theta(\cdot|p,q)$ using a separate BERT

- Model trained by cross-entropy loss, at test time predict span with highest probability
- What if there are multiple gold spans in a passpage? Can optimize marginal log likelihood $\log \sum_{(i,j) \in S(p,q)} p_\theta(i,j|p,q)$
- Can marginalize over passages, different probabilistic assumptions yield different results (Cheng et al., 2020)

Open-Domain Question Answering

- Original SQuAD-style QA (Rajpurkar et al., 2016): Supplies (p, q) where passage p contains an answer phrase to question q
  - Can just train a span-selection model, no need to consult a KB
- **Open-domain QA**: Only supplies q, but assumes a KB of passages $p_1 \ldots p_M$ (e.g., Wikipedia articles broken into 512-long text blocks)
- Typical pipeline approach
  1. Use efficient **score**(q, p) to retrieve top-K passages: TFIDF, BM25, dual encoder
  2. Apply span-selection model (multi-paragraph version)



(Chen et al., 2017)

Fact Checking

- Example: FEVER (Thorn et al., 2018)
  - Dataset of $185k$ claims manually labed as supported (S), refuted (R), or neutral (N)
  - S/R claims: Additionally have *evidence sentences* from Wikipedia articles
- A claim might require multiple evidence sentences (from multiple articles)
  - **Claim.** Giada at Home was only available on DVD
  - E1. "It first aired on October 18, 2008 on the Food Network." (2nd sent on *Giada at Home*)
  - E2. "Food Network is an American basic cable and satellite television channel." (1st sent on *Food Network*)
  - **Label.** R
- **FEVER Accuracy.** Prediction correct iff label correct **AND** predicted evidence set (at most 5 sents) cover annotated evidence set
  - Only 32% with a simple pipeline: (1) Retrieve top-5 articles by TFIDF, (2) Retrieve top-5 sentences by TFIDF, (3) Apply an NLI model on evidence(concat)-claim pairs

Retrieval-Based Dialogue

- Example: Wizard of Wikipedia (WoW) (Dinan et al., 2019)
  - Dataset of $20k$ dialogues (total $200k$ utterances)
  - Each dialogue focuses on one (or more) of $> 1k$ diverse *topics* (e.g., commuting, Gouda cheese, music festivals, podcasts, bowling)
  - Assymetric agents: **Wizard** sees Wiki articles on that topic, **Apprentice** does not (during annotation)
  - Goal: Model Wizard
- At every turn, the system (1) selects a relevant sentence from Wikipedia ($93m$ sents) and (2) predicts an utterance conditioning on that sentence
  - Original paper: Candidate sentences $c = \{c_1 \dots c_K\}$ from top-7 articles (TFIDF using 1st and last two utterances)
  - **Retrieval** model (nonparametric): Return training response r with highest $\textbf{score}_\theta(\text{history}, c, r)$
  - **Generative** model: Generate from a seq2seq model conditioning on history and c
- Evaluation: Unigram F1, perplexity (against annotated data)

Example from WoW

History

- **Wizard**: Obesity is a condition where excess body fat is very high
- **Apprentice**: That's true, I think obesity is a big problem in the US right now.
- **Wizard**: Yes and the negative health effects can include mortality
- **Apprentice**: Definitely, it's a big problem. Do you know how common obesity is in the US right now?
- **Wizard**: 1 in 4 and the causes are varied from over eating to lack of activity and of course genetics play a part
- **Apprentice**: Wow that's crazy it's so common. Are there any cures for obesity?

Gold response

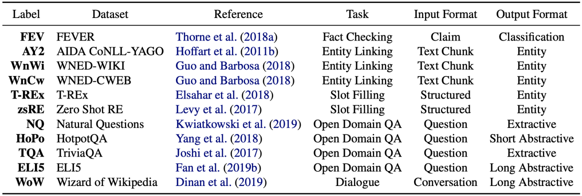
- **Wizard**: You can try changes to diet and exercising to increase muscular gain.
- **Wipepedia article**: *Obesity*: Obesity is mostly preventable through a combination of social changes and personal choices. Changes to diet and exercising are the main treatments. Diet quality can be improved by . . .

Relation Extraction and Slot Filling

- **Relation extraction (RE)**
  - Task of identifying instances of relations (e.g., $\texttt{nationality}(\texttt{person}, \texttt{country})$) in passages of natural text
  - Predefined set of relation types R (e.g., `educated-at`, `occupation`, etc., drawn from Wikidata)
  - Annotation: Passage p, two spans (x, y) representing predicate-argument, their relation $r(x, y|p) \in R$
  - Pipeline approach: Find spans (e.g., by NER), train a relation classifier
- **Slot filling**
  - Input: $r(x, \cdot|p)$ where $r \in R$, x is some entity in KB, p passage
  - Output: String y such that $r(x, y)$ is true based on p
  - Can be solved as QA: $\texttt{educated-at}(Einstein, \cdot|p)$ is asking "Where did Einstein study?" with passage p, return answer span (Levy et al., 2017)
- Open-domain slot filling
  - System doesn't get p, given $r(x, \cdot)$ must retrieve passages from which y can be extracted
  - Analogous to open-domain QA

Benchmarks for Knowledge-Intensive Language Tasks

- Example: KILT (Petroni et al., 2021)
- Wikipedia-based tasks a uniform framework



Label	Dataset	Reference	Task	Input Format	Output Format
FEV	FEVER	Thorne et al. (2018a)	Fact Checking	Claim	Classification
AY2	AIDA CoNLL-YAGO	Hoffart et al. (2011b)	Entity Linking	Text Chunk	Entity
WnWi	WNED-WIKI	Guo and Barbosa (2018)	Entity Linking	Text Chunk	Entity
WnCw	WNED-CWEB	Guo and Barbosa (2018)	Entity Linking	Text Chunk	Entity
T-REx	T-REx	Elsahar et al. (2018)	Slot Filling	Structured	Entity
zsRE	Zero Shot RE	Levy et al. (2017)	Slot Filling	Structured	Entity
NQ	Natural Questions	Kwiatkowski et al. (2019)	Open Domain QA	Question	Extractive
HoPo	HotpotQA	Yang et al. (2018)	Open Domain QA	Question	Short Abstractive
TQA	TriviaQA	Joshi et al. (2017)	Open Domain QA	Question	Extractive
ELI5	ELI5	Fan et al. (2019b)	Open Domain QA	Question	Long Abstractive
WoW	Wizard of Wikipedia	Dinan et al. (2019)	Dialogue	Conversation	Long Abstractive

- Every example has an associated text span in Wikipedia ("provenance", whole article for EL)
- Main challenge: Different datasets used different versions of Wikipedia dump, resolved by careful re-mapping
- Different datasets have different evaluation metrics: accuracy (EL, fact checking, slot filling), exact match (QA), Rouge-L (abstractive generation), unigram-F1 (WoW)
  - "KILT score": Only award points if the gold provenance is ranked at the top by the system