Autoregressive Models for Retrieval

Kavin R V (19163)
Dr. Maunendra Sankar Desarkar (IIT Hyderabad)

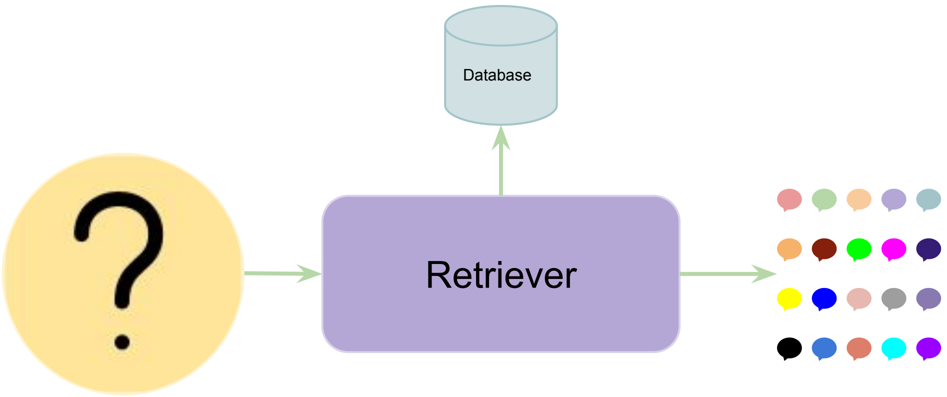
Content

- Background
- Foresight
- Intent Detection
- NCI Contra
- Conclusion and Future Work

Background

- Information Retrieval
- Sparse Retrieval
- Dense Retrieval
- Generative Retrieval

Information retrieval



Sparse Retrieval

This is a query

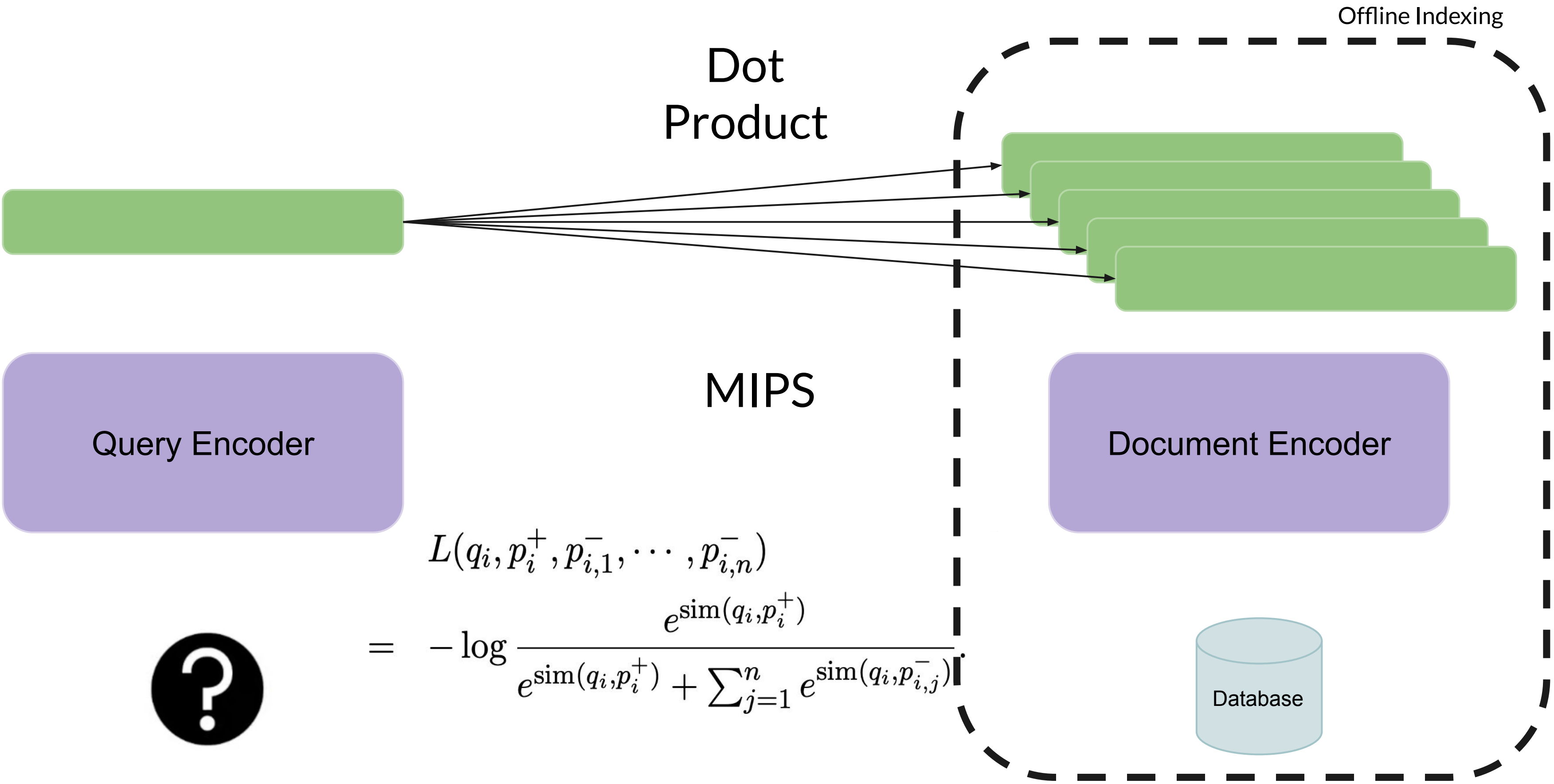
Vocab Size

1	0	0	1	...	1	0	1	0	...	0

0	0	1	1	...	0	0	0	0	...	1
1	0	0	1	...	1	0	1	0	...	0
1	0	0	1	...	1	0	1	0	...	0
1	0	0	1	...	1	0	1	0	...	0
1	0	0	1	...	1	0	1	0	...	0
1	0	0	1	...	1	0	1	0	...	0
1	0	0	1	...	1	0	1	0	...	0

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

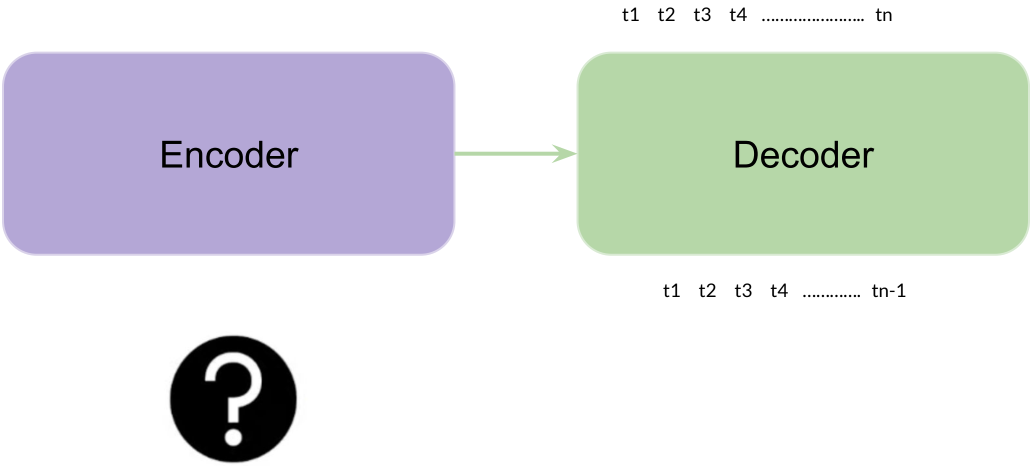
Information Retrieval: Dense Retrieval



$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) = -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, p_{i,j}^-)}}.$$

Generative Retrieval

t1 t2 t3 t4 …………………. tn = Document Identifier



Generative Retrieval (Token Based)

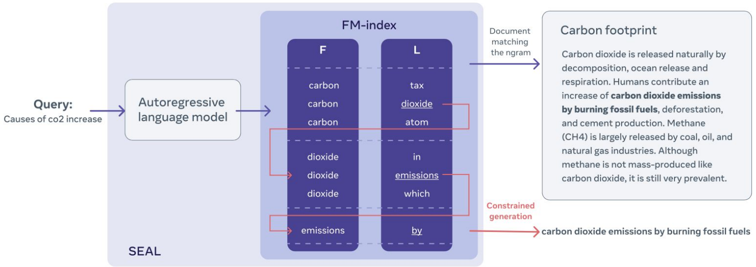


Figure 1: High-level SEAL architecture, composed of an autoregressive LM paired with an FM-Index, for which we show the first (F) and last (L) columns of the underlying matrix (more details in Sec 3.1). The FM-index constraints the autoregressive generation (*e.g.*, after *carbon* the model is contrained to generate either *tax*, *dioxide* or *atom* in the example) and provides the documents matching (*i.e.*, containing) the generated ngram (at each decoding step).

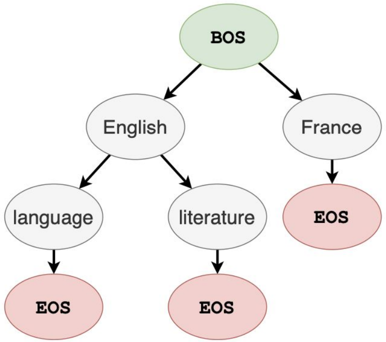
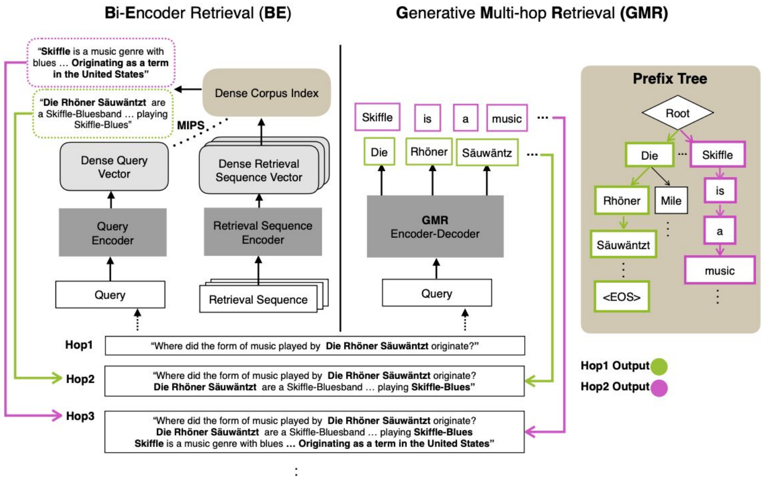
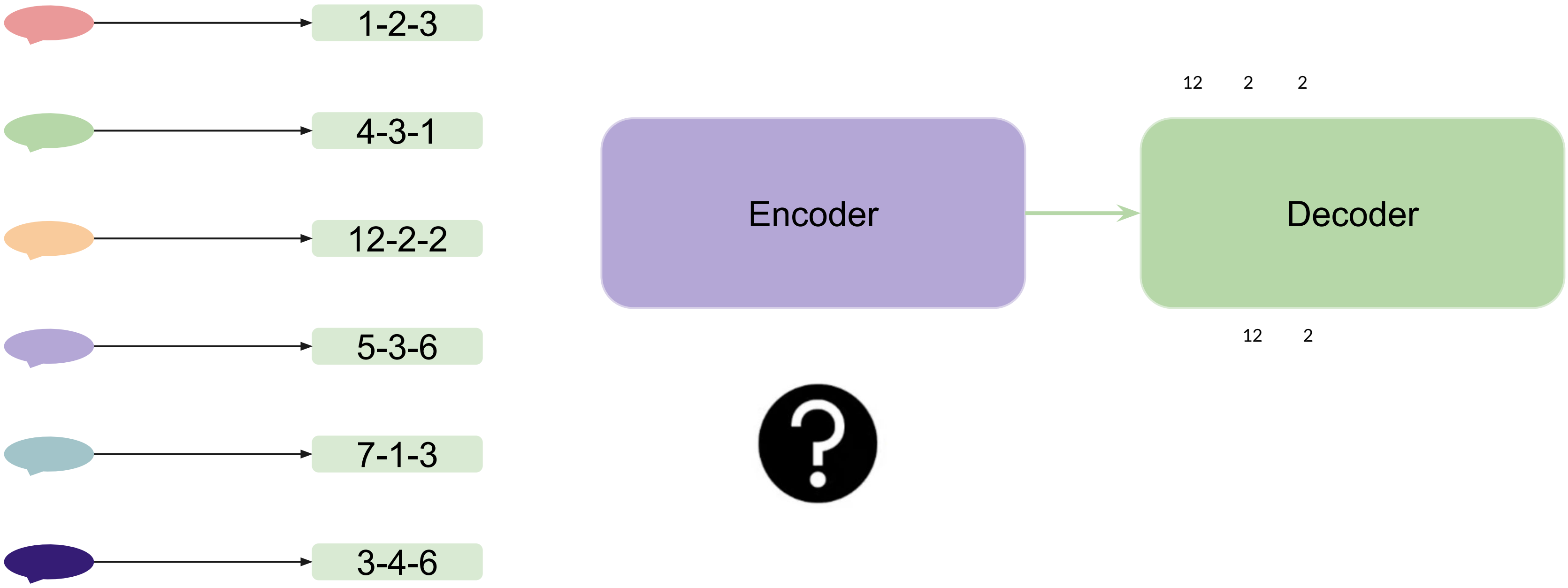


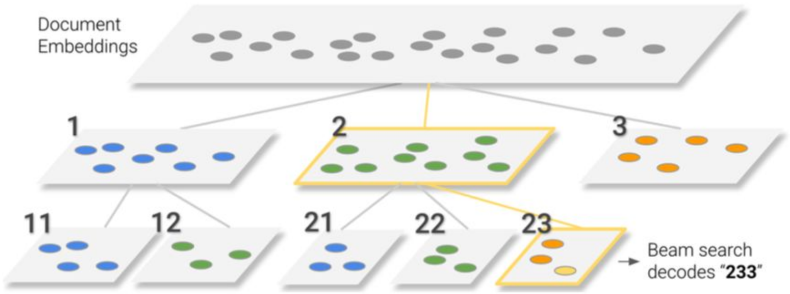
Figure 9: Example of prefix tree (trie) structure where the allowed entities identifiers are 'English language', 'English literature' and 'France'. Note that at the root there is the start-of-sequence token SOS and all leaves are end-of-sequence tokens EOS. Since more that one sequence has the same prefix (i.e., 'English'), this end up being an internal node where branches are the possible continuations.

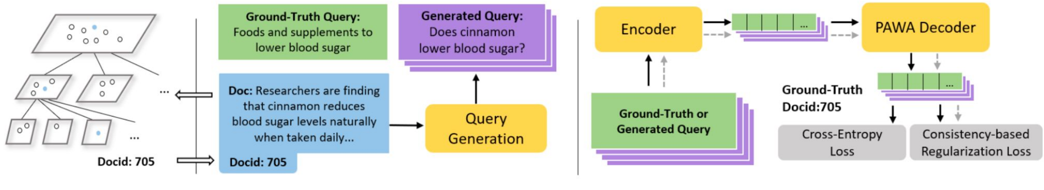


Docid

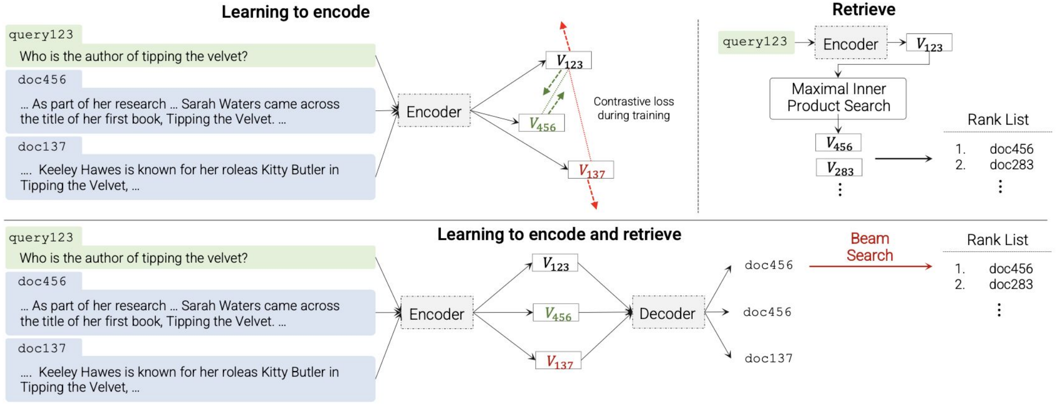


Generative Retrieval (Docid Based)

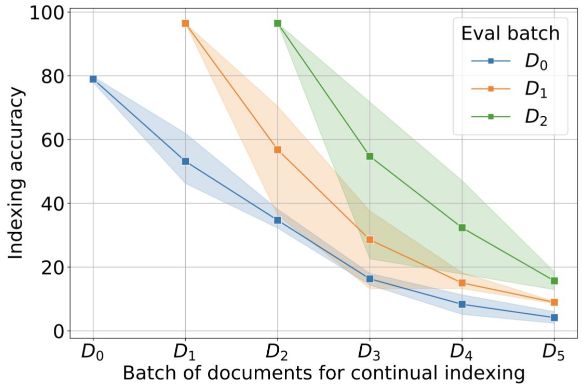


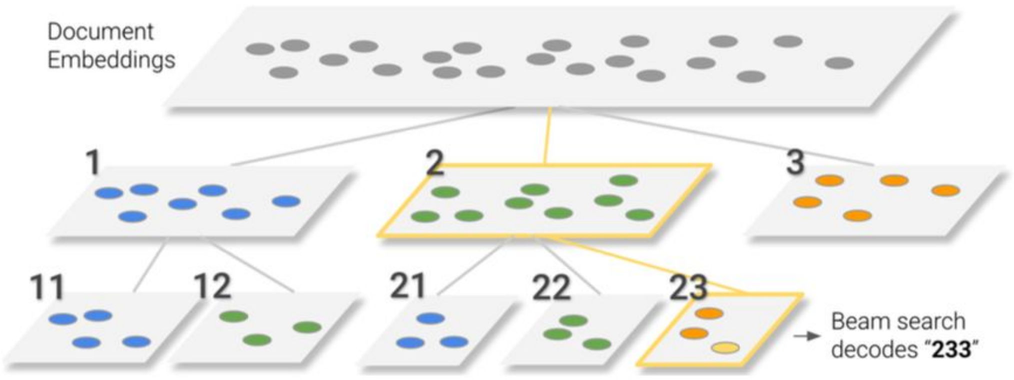


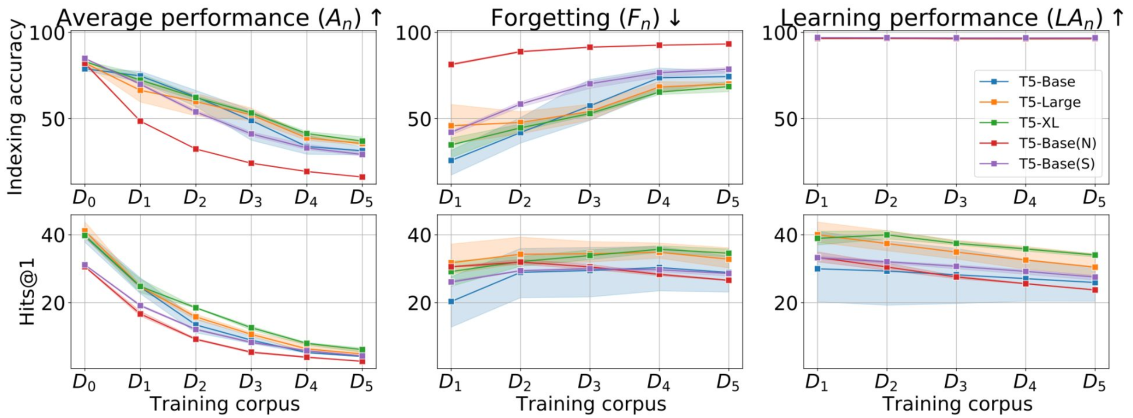
(a) Preprocessing (b) Query Generation (c) Training pipeline of Neural Corpus Indexer



Updating Generative Retrieval







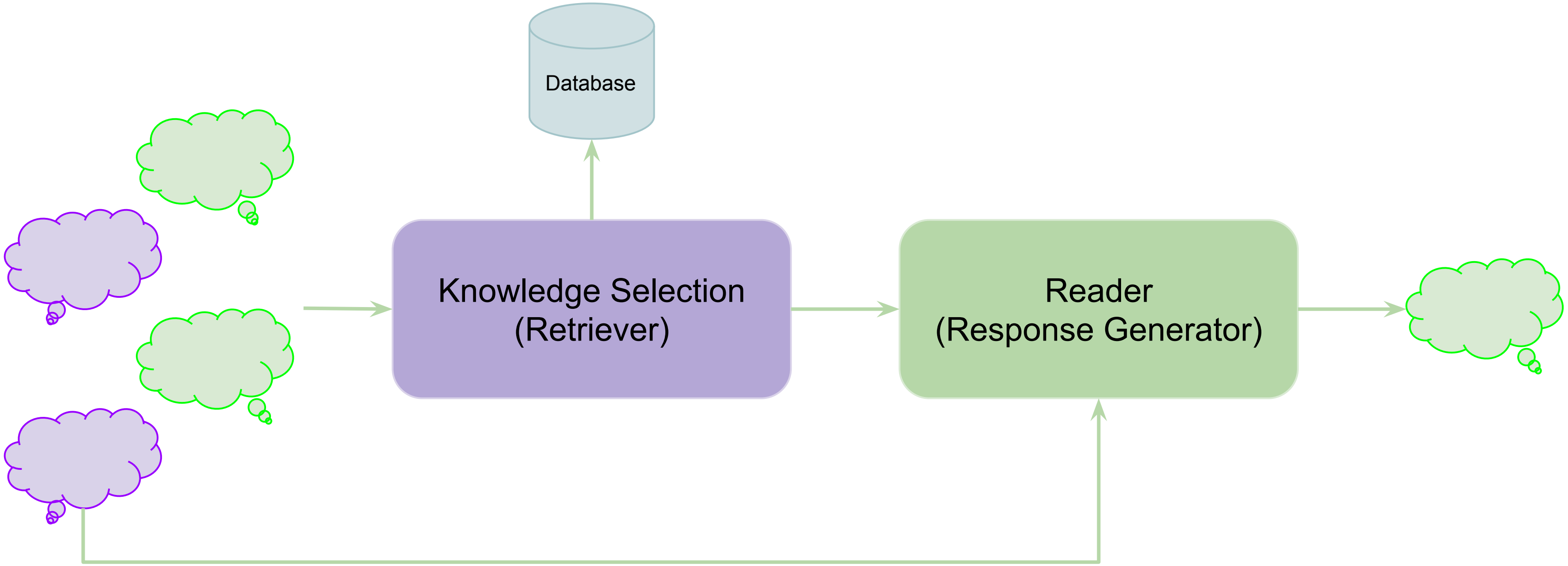
Foresight

- Background
- Methodology
- Experiments and Results

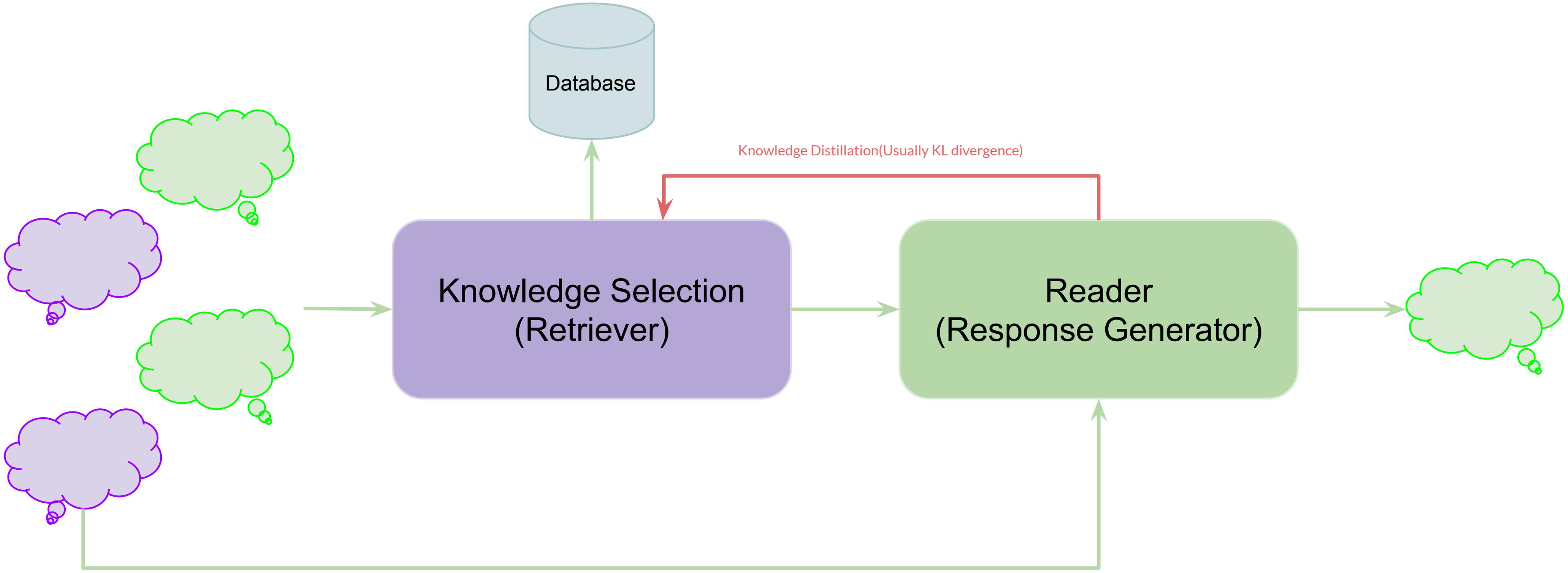
Background-Dialogue Systems

- Knowledge Grounded Conversation
- Knowledge Distillation
- Re-Ranking
- RankT5

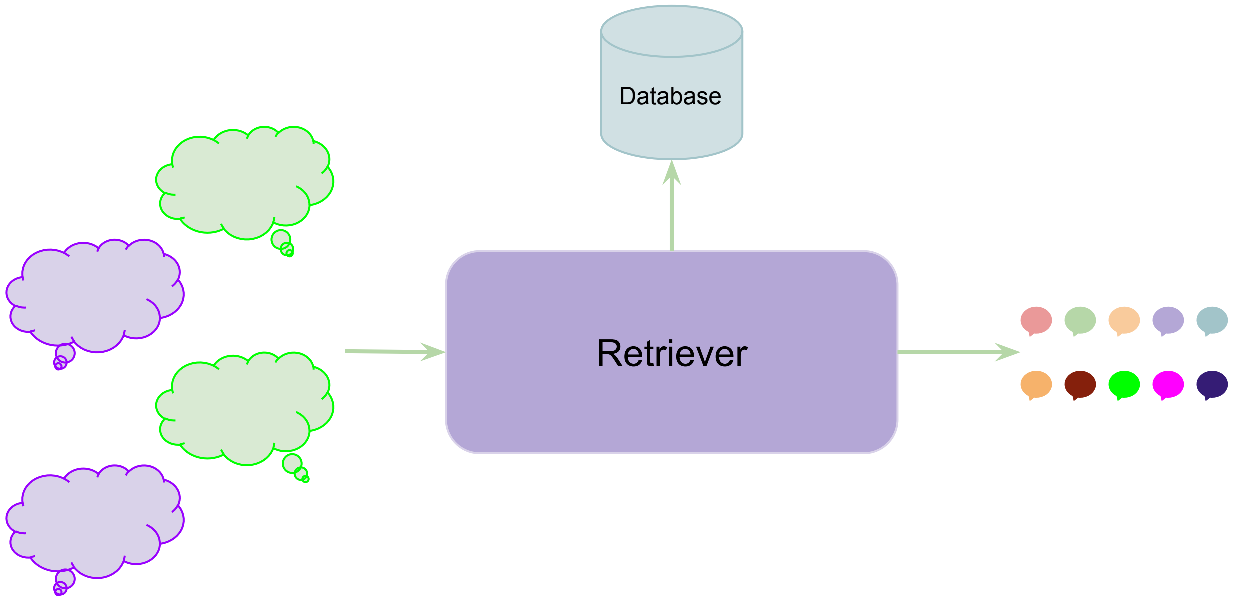
Knowledge Grounded Conversation



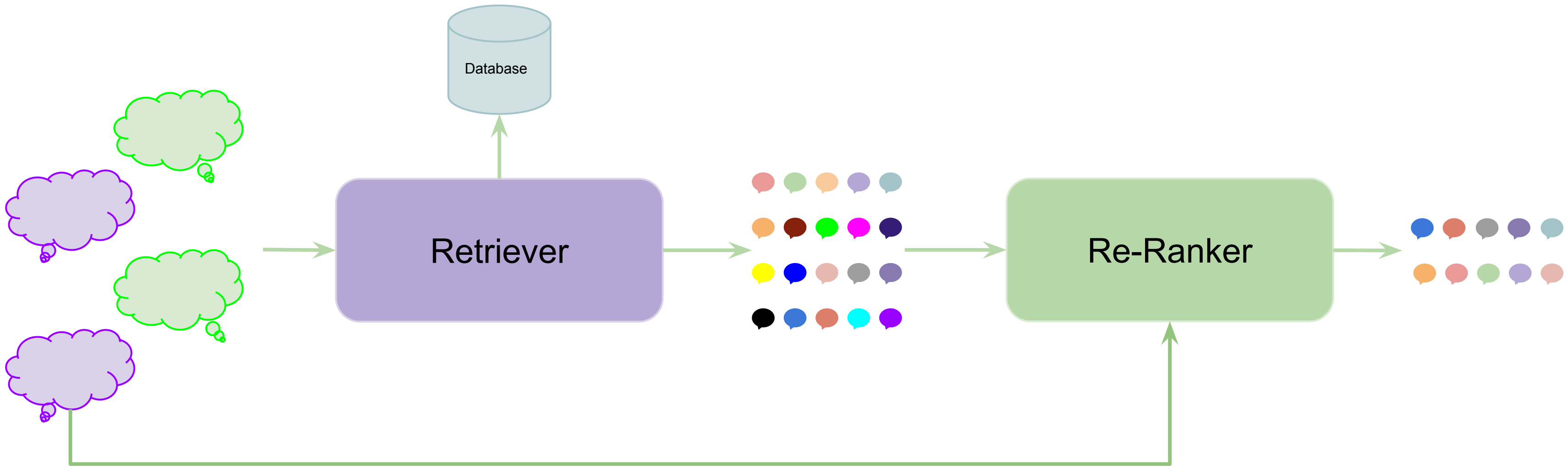
Knowledge Distillation



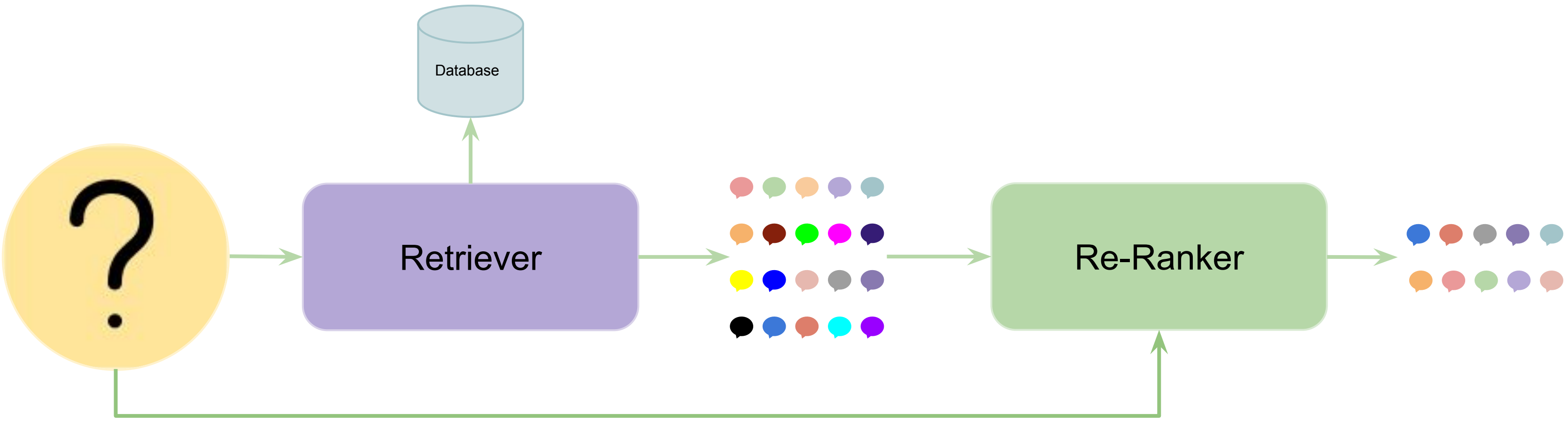
Re-Ranking



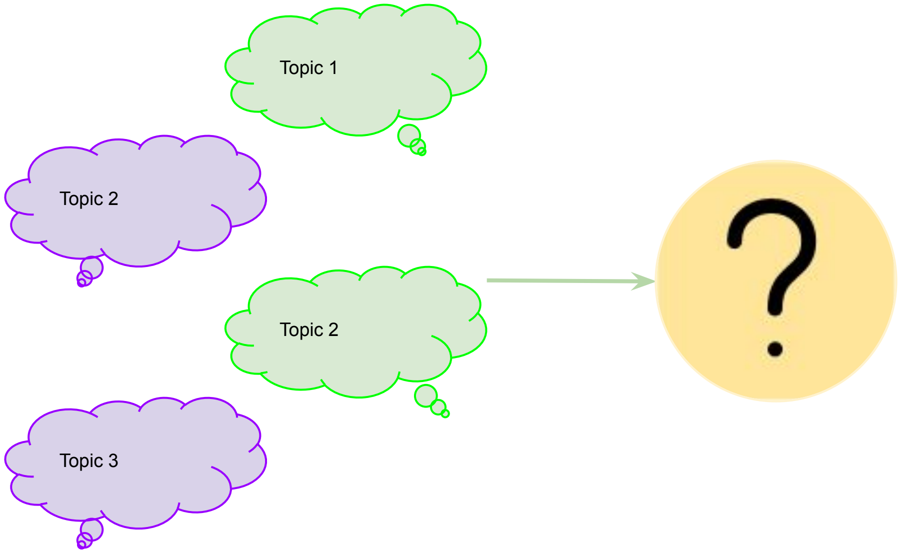
Re-Ranking



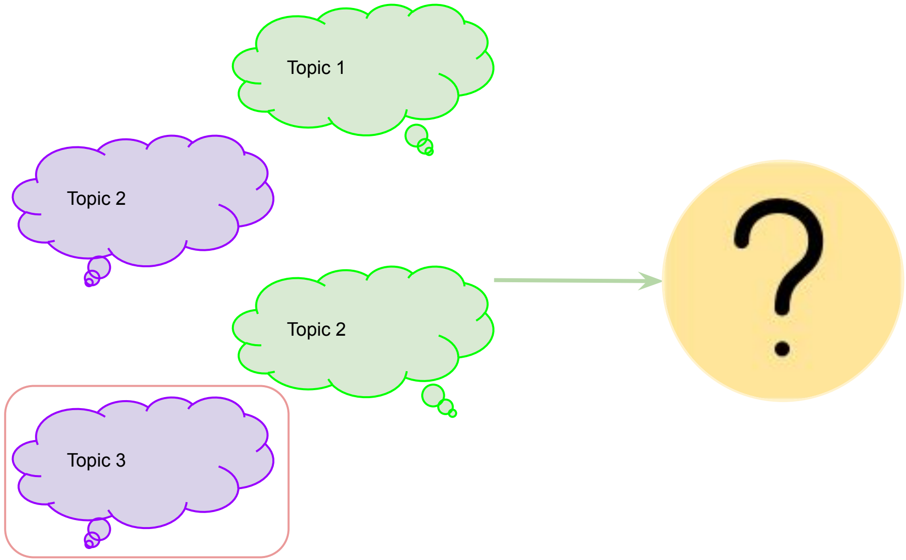
Query For Conversations



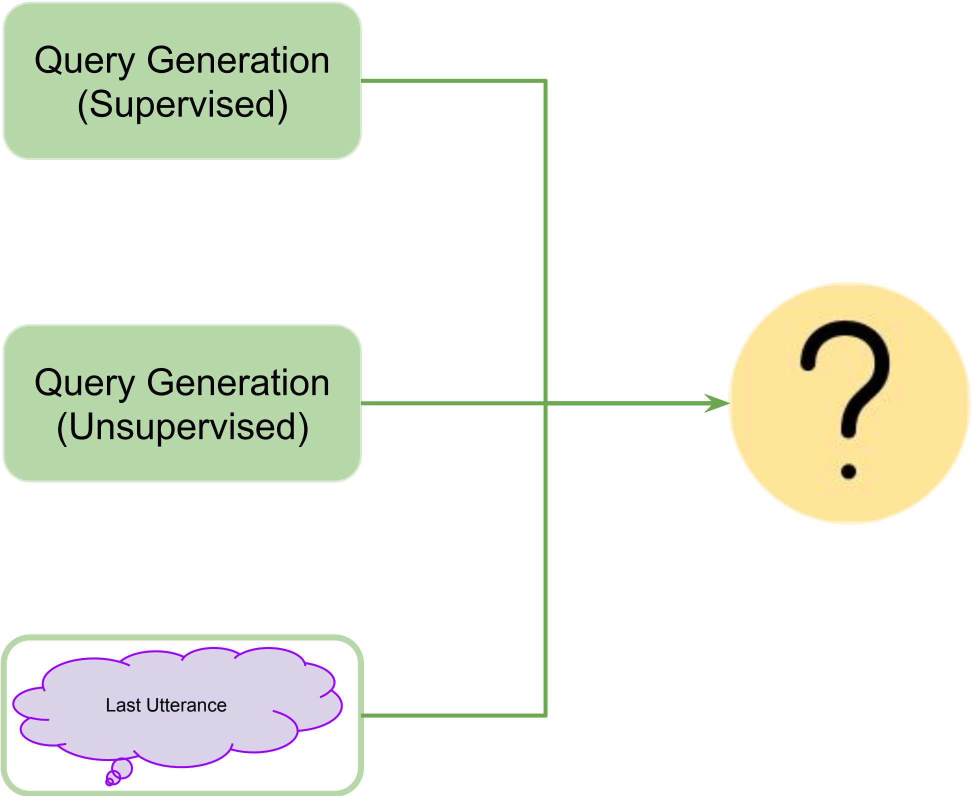
Query For Conversations



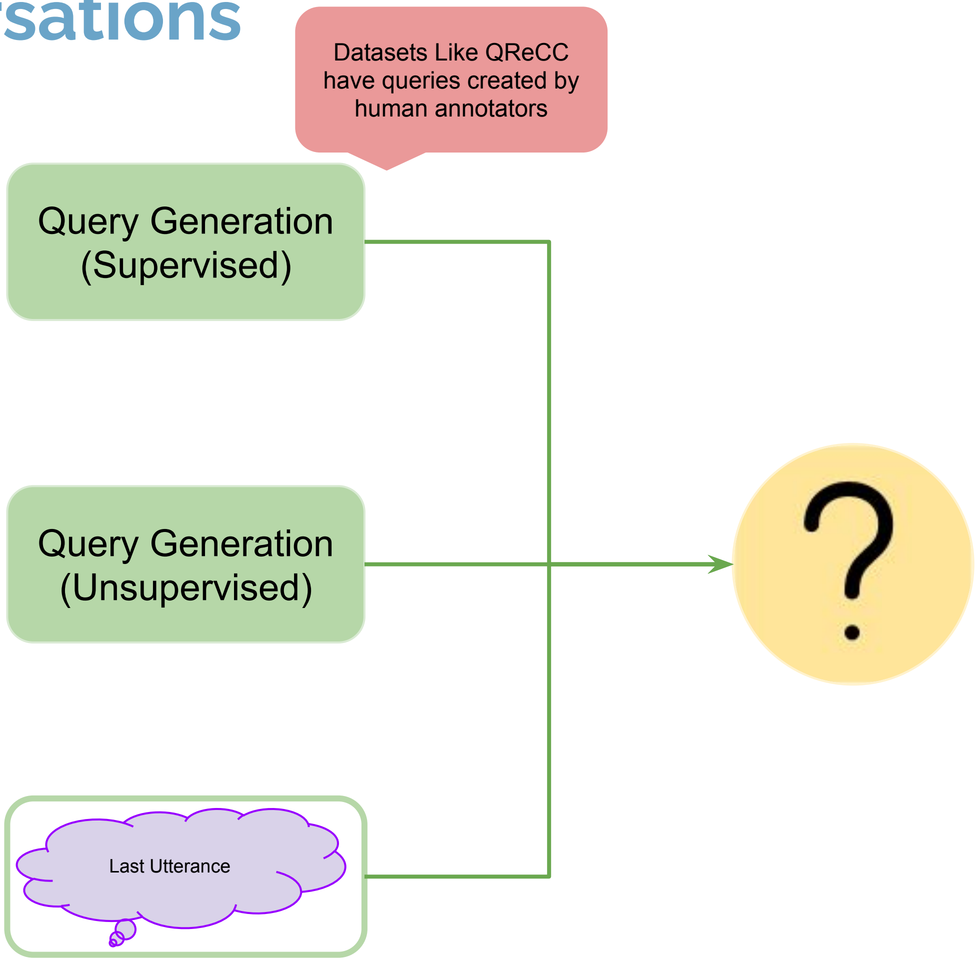
Query For Conversations



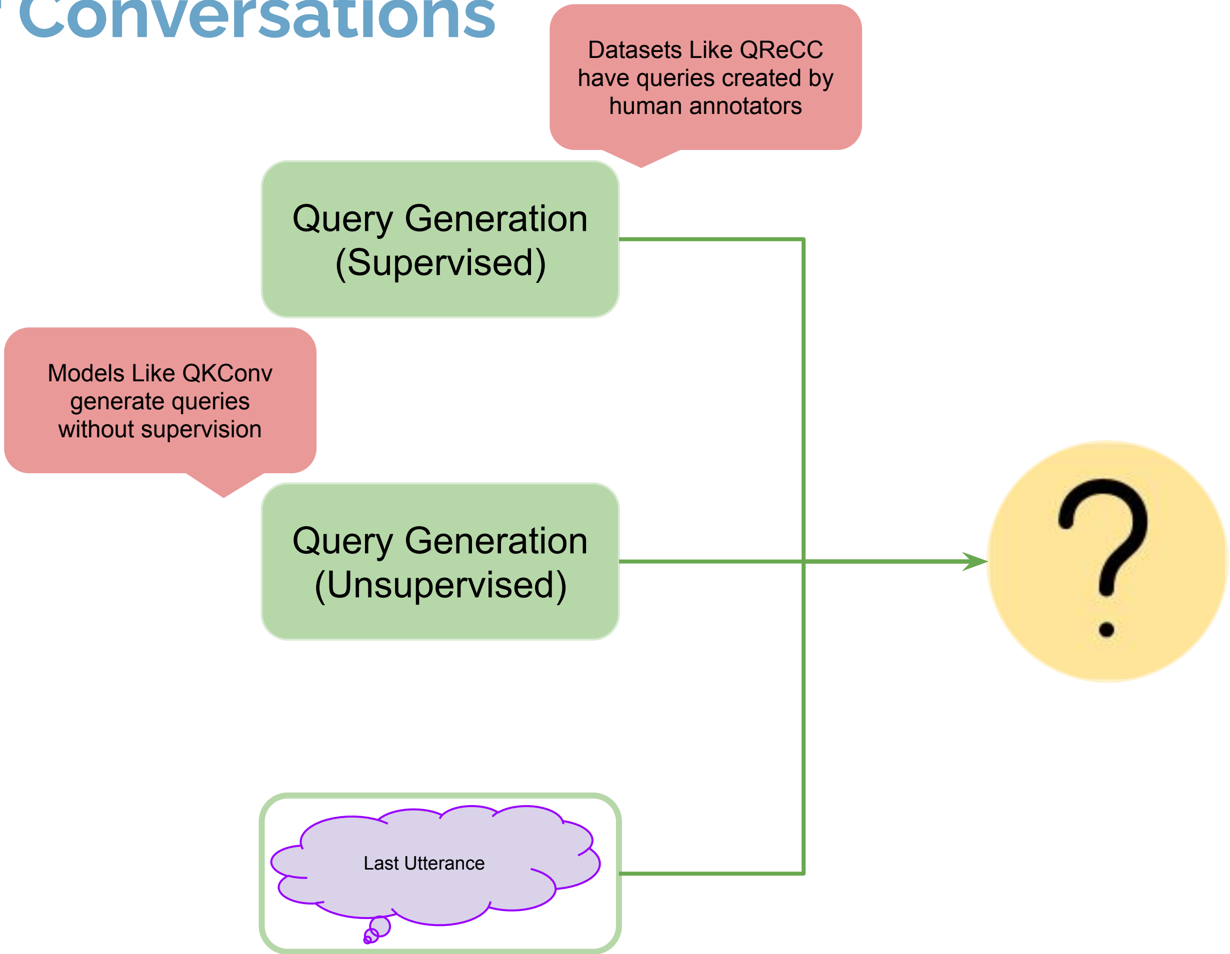
Query For Conversations



Query For Conversations



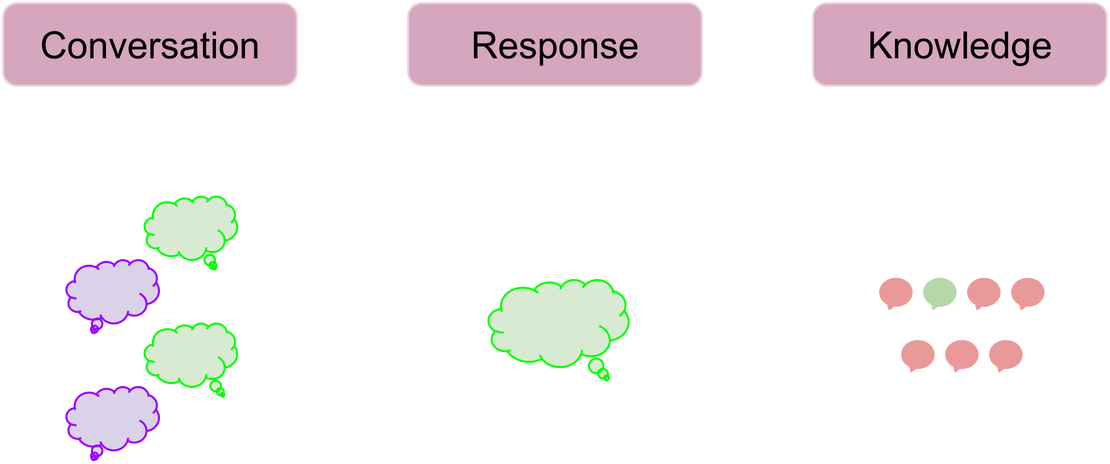
Query For Conversations



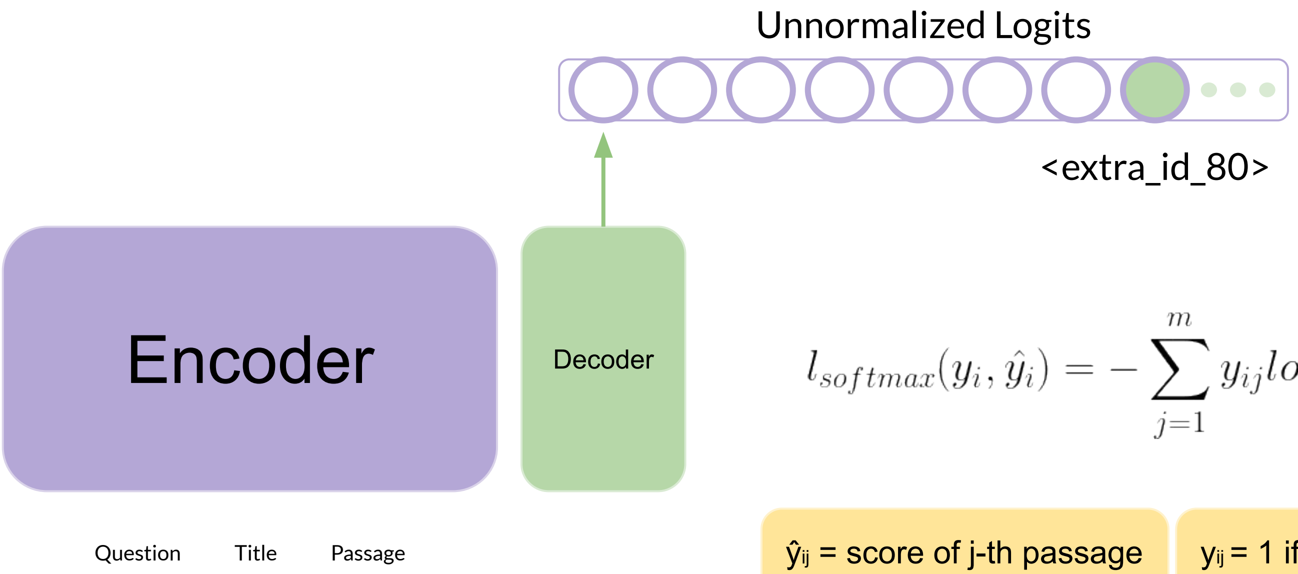
Methodology

- Wizard of Wikipedia
- RankT5
- Masked Response
- Keyword Estimation
- Generated Masked Response

Wizard of Wikipedia



Rank-T5



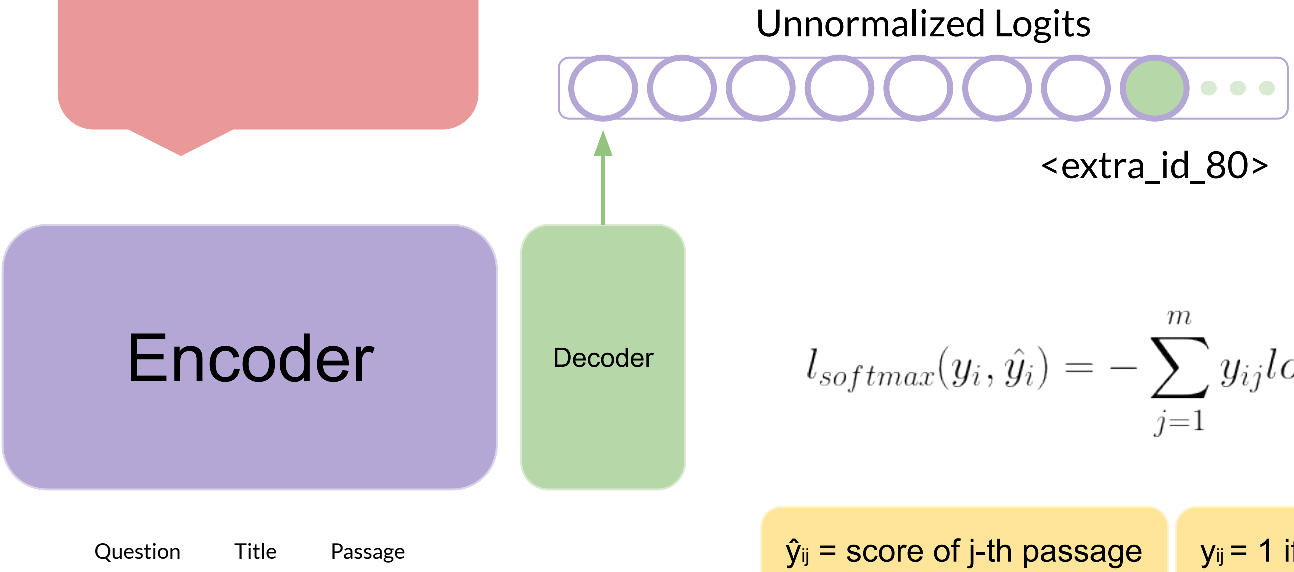
$$l_{softmax}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} log \left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}} \right)$$

$\hat{y}_{ij}$ = score of j-th passage for i-th query

y_{ij} = 1 if the passage j is a provenance to query i

Rank-T5

→ Good Performance Compared to other Re-Rankers

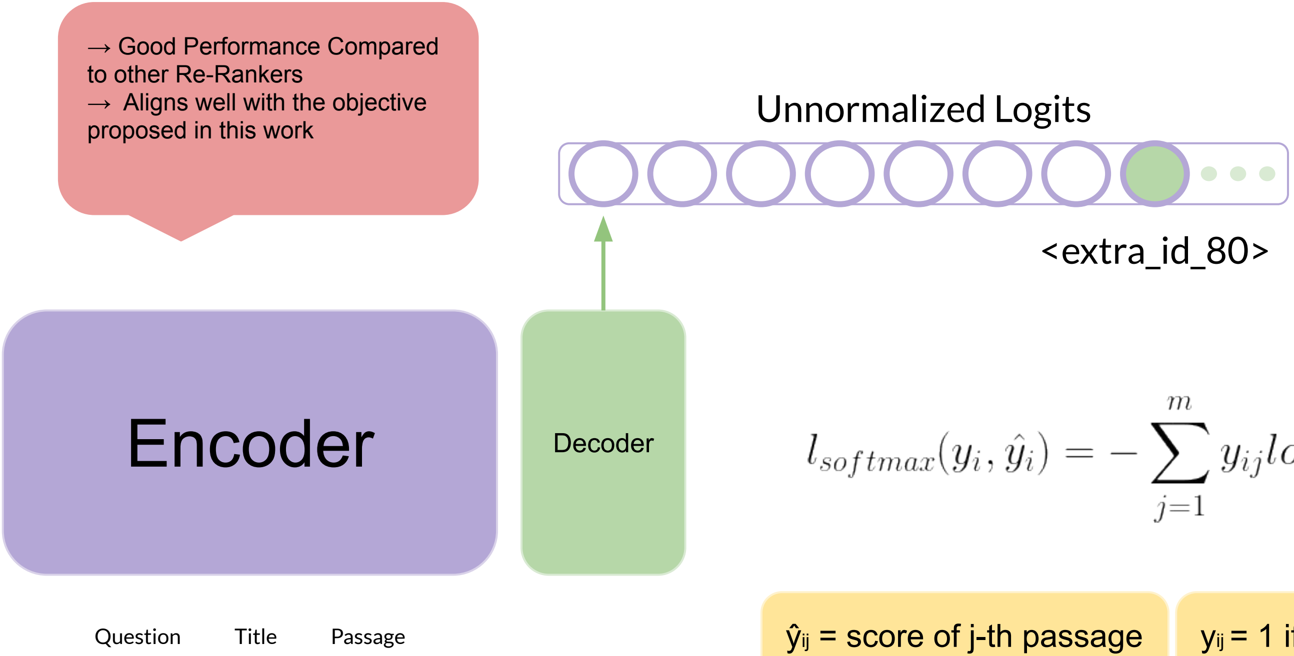


$$l_{softmax}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} log \left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}} \right)$$

$\hat{y}_{ij}$ = score of j-th passage for i-th query

y_{ij} = 1 if the passage j is a provenance to query i

Rank-T5

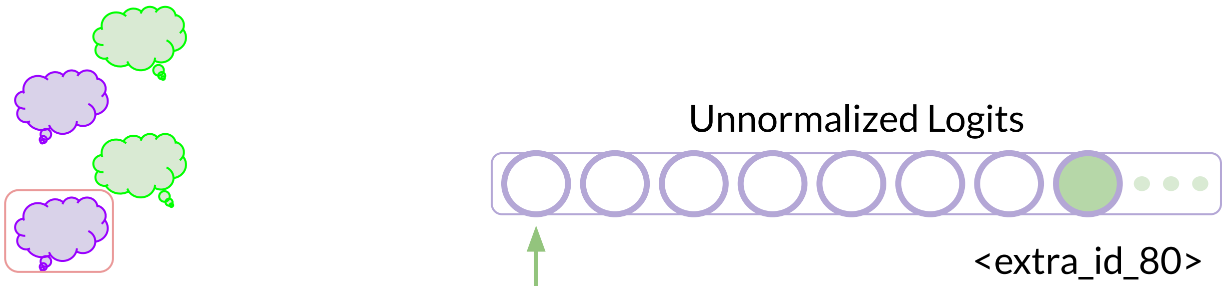


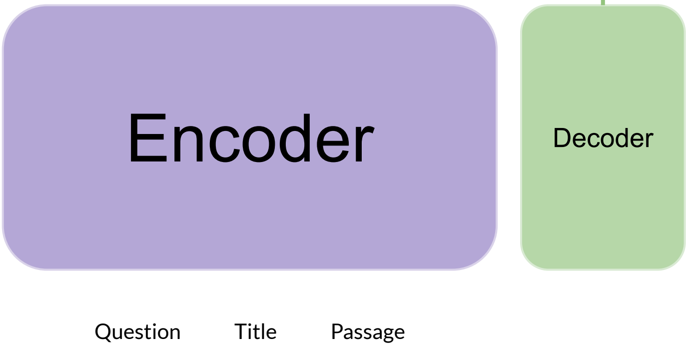
$$l_{softmax}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} log \left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}} \right)$$

$\hat{y}_{ij}$ = score of j-th passage for i-th query

y_{ij} = 1 if the passage j is a provenance to query i

Rank-T5





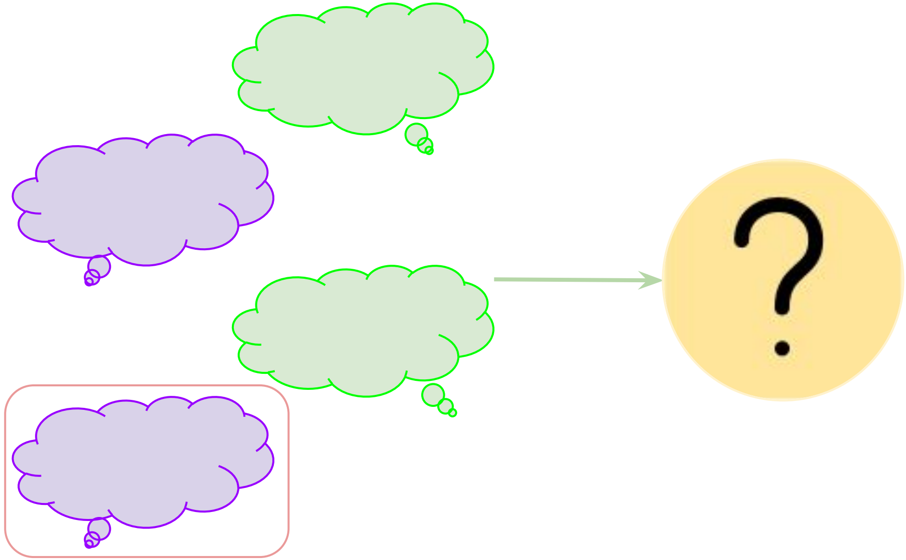
$$l_{softmax}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} log\left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}}\right)$$

$\hat{y}_{ij}$ = score of j-th passage for i-th query

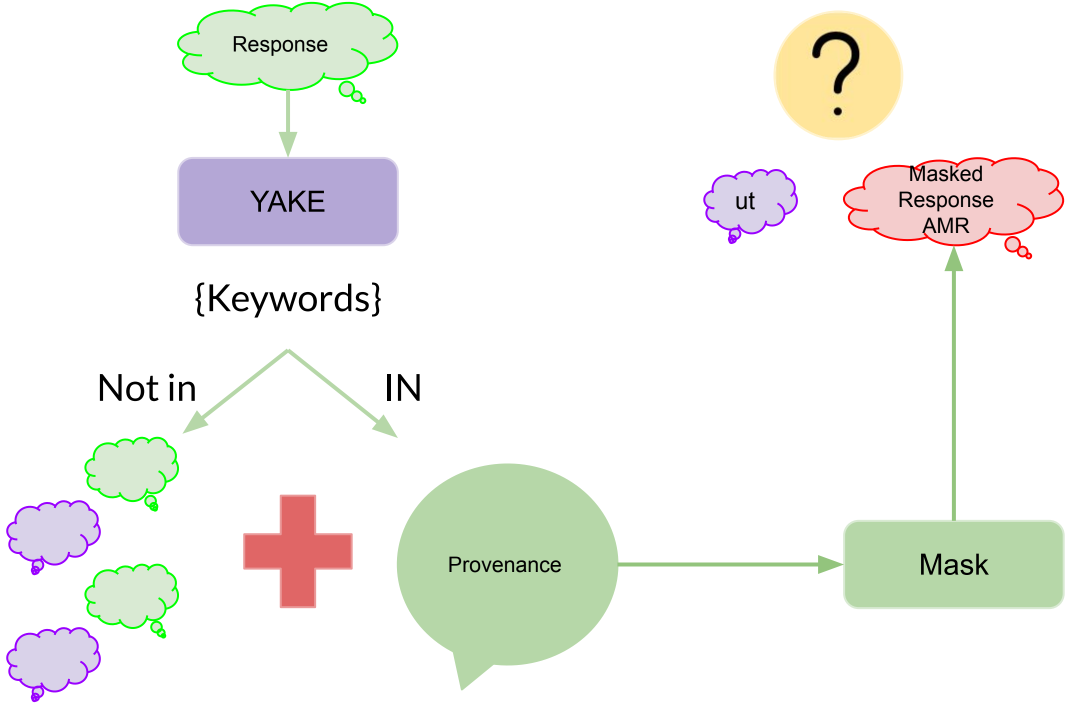
y_{ij} = 1 if the passage j is a provenance to query i

Rank-T5

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 query as last utterance	88.57	81.12	91.59	95.29	97.20	98.75	81.12	87.72	89.58	90.40	91.00



Masked Response



Masked Response

Encoder

Decoder

question: Masked Response   title: Passage Title   context: Passage

How does social interaction relate to transition as mentioned? <eou> Kids <extra_id_3> to interact with their peers.Record shows that the first kindergarten centers were opened late 18th <extra_id_2> in <extra_id_1> and <extra_id_0>

Kids **learn** to interact with their peers.Record shows that the first kindergarten centers were opened late 18th **century** in **Bavaria** and **Strasbourg**

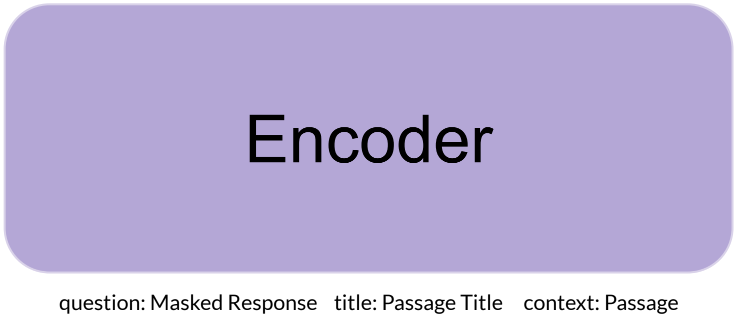
Masked Response

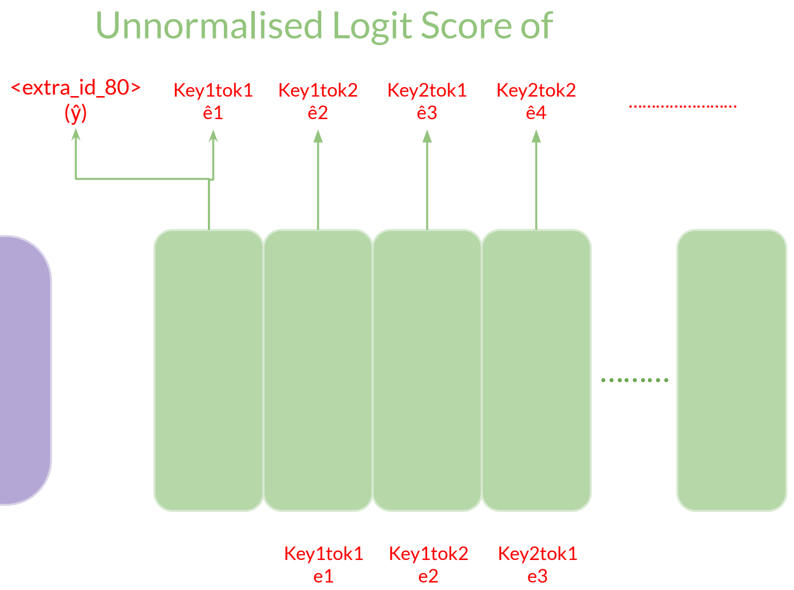
question: How does social interaction relate to transition as mentioned? <eou> Kids***<extra_id_3>*** to interact with their peers.Record shows that the first kindergarten centers were opened late 18th***<extra_id_2>*** in***<extra_id_1>*** and***<extra_id_0>*** **title**: Kindergarten **passage**: Kindergarten (; from German, which literally means "garden for the children") is a preschool educational approach traditionally based on playing, singing, practical activities such as drawing, and social interaction as part of the transition from home to school. At first such institutions were created in the late 18th **century** in **Bavaria** and **Strasbourg** to serve children whose parents both worked out of the home. The term was coined by the German Friedrich Fröbel, whose approach globally influenced early-years education. Today, the term is used in many countries to describe a variety of educational institutions and **learn**ing spaces for children ranging from two to seven years of age, based on a variety of teaching methods. In 1779, Johann Friedrich Oberlin and Louise Scheppler founded in Strasbourg an early establishment for caring for and educating pre-school children whose parents were absent during the day. At about the same time, in 1780, similar infant establishments were established in Bavaria. In 1802, Princess Pauline zur Lippe established a preschool center in Detmold, the capital of the then principality of Lippe, Germany (now in the State of North Rhine-Westphalia). In 1816, Robert Owen, a philosopher and pedagogue, opened the first British and probably globally the first infants school in New Lanark, Scotland.</s>

Keyword Estimation

$$\mathrm{RankScore}(q_i, p_j) = \hat{y}_{ij}$$

$$\mathrm{KEScore}(q_i, p_j) = \sum_k \hat{e}_{ik} = \hat{z}_{ij}$$





Keyword Estimation

$$l_{softmax}(y_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} log\left(\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}}\right)$$

$$l_{softmax}(y_i, \hat{z}_i) = -\sum_{j=1}^{m} y_{ij} log\left(\frac{e^{\hat{z}_{ij}}}{\sum_{j'} e^{\hat{z}_{ij'}}}\right)$$

$$l_{kl}(\hat{Z}||\hat{Y}) = \sum_{j=1}^{m} \frac{e^{\hat{z}_{ij}/\tau}}{\sum_{j'} e^{\hat{z}_{ij'}/\tau}} log\left(\frac{\frac{e^{\hat{y}_{ij}}}{\sum_{j'} e^{\hat{y}_{ij'}}}}{\frac{e^{\hat{z}_{ij}/\tau}}{\sum_{j'} e^{\hat{z}_{ij'}/\tau}}}\right)$$

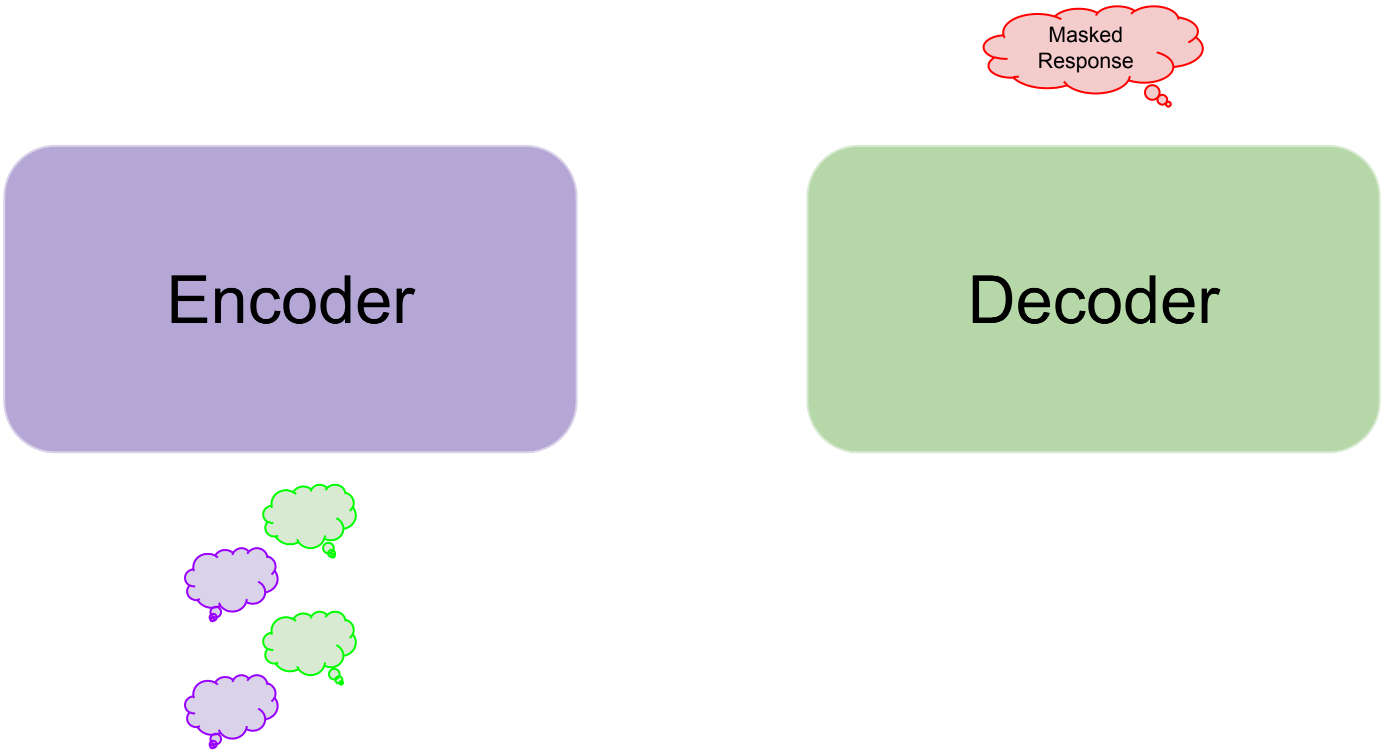
$$l_{SKL} = l_{KL}(\text{stopgrad}(\hat{Z})||\hat{Y}) + l_{KL}(\text{stopgrad}(\hat{Y})||\hat{Z})$$

$$l_1 = l_{softmax}(y_i, \hat{y}_i) + l_{softmax}(y_i, \hat{z}_i)$$

$$l_2 = l_{softmax}(y_i, \hat{y}_i) + l_{KL}(\hat{Z}||\hat{Y})$$

$$l_3 = l_{softmax}(y_i, \hat{y}_i) + \lambda \cdot l_{SKL}$$

Generated Masked Response



Experiments and Results

- Train = AMR, Test = AMR
- Train = AMR, Test = GMR
- Train = GMR, Test = GMR
- Ablation Study

Results (test_q = ut+amr, train_q = ut+amr)

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 query as last utterance	88.57	81.12	91.59	95.29	97.20	98.75	81.12	87.72	89.58	90.40	91.00
Rank-T5 w/o KE loss	94.79	90.94	97.09	98.39	99.18	99.59	90.94	94.82	95.47	95.81	95.97
Rank-T5 With L1 loss	94.79	91.00	96.93	98.39	98.99	99.59	90.99	94.74	95.47	95.73	95.96
Rank-T5 L2 loss (t=2)	94.65	90.72	96.90	98.42	99.10	99.56	90.72	94.62	95.38	95.67	95.85
Rank-T5 L2 loss (t=3)	94.96	91.29	96.98	98.56	99.16	99.83	91.29	94.88	95.67	95.93	96.18
Rank-T5 L2 loss (t=5)	94.40	90.34	96.69	98.23	99.08	99.37	90.34	94.34	95.12	95.48	95.60
Rank-T5 L3 loss (t=1)	**95.46**	**91.53**	**97.41**	**98.55**	**99.28**	**99.84**	**91.53**	**95.61**	**95.96**	**96.07**	**96.90**

Results (test_q = ut+gmr, train_q = ut+amr)

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 query as last utterance	88.57	81.12	91.59	95.29	97.20	98.75	81.12	87.72	89.58	90.40	91.00
Rank-T5 w/o KE loss	85.29	76.14	88.08	93.33	96.60	98.23	76.14	83.68	86.30	87.71	88.34
Rank-T5 With L1 loss	85.65	76.41	89.04	93.99	96.76	98.39	76.41	84.38	86.85	88.05	88.68
Rank-T5 L2 loss (t=2)	86.93	78.45	90.21	94.50	97.03	98.45	78.45	85.87	88.01	89.11	89.66
Rank-T5 L2 loss (t=3)	87.02	78.62	90.32	94.53	96.98	98.45	78.62	86.00	88.11	89.16	89.73
Rank-T5 L2 loss (t=5)	86.74	78.12	90.04	94.56	97.03	98.56	78.12	85.64	87.90	88.97	89.56
Rank-T5 L3 loss (t=1)	87.23	78.94	90.42	94.70	97.17	98.56	78.94	86.19	88.32	89.39	89.92

Results (test_q = ut+gmr, train_q = ut+gmr)

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 query as last utterance	88.57	81.12	91.59	95.29	97.20	98.75	81.12	87.72	89.58	90.40	91.00
Rank-T5 w/o KE loss	89.69	83.00	92.13	95.78	**97.88**	**99.05**	83.00	88.76	90.59	91.49	**91.94**
Rank-T5 With L1 loss	89.30	82.18	92.19	96.06	97.71	98.91	82.18	88.50	90.43	91.14	91.61
Rank-T5 L2 loss (t=2)	89.52	82.81	91.78	95.73	97.77	98.97	82.81	88.47	90.44	91.32	91.79
Rank-T5 L2 loss (t=3)	89.44	82.54	91.94	96.06	97.82	98.94	82.54	88.47	90.53	91.29	91.72
Rank-T5 L2 loss (t=5)	89.56	82.59	92.36	96.03	98.07	99.13	82.59	88.75	90.59	91.47	91.88
Rank-T5 L3 loss (t=1)	**89.82**	**83.92**	**92.38**	**96.19**	97.93	98.88	**83.92**	**88.89**	**90.79**	**91.54**	91.91

Results {Ablation Study} (q = gmr)

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 query as last utterance	88.57	81.12	91.59	95.29	97.20	98.75	81.12	87.72	89.58	90.40	91.00
Rank-T5 w/o KE loss	82.86	72.58	85.58	91.78	95.13	97.50	72.58	80.78	83.88	85.33	86.24
Rank-T5 With L1 loss	83.22	73.04	86.32	91.78	95.24	97.61	73.04	81.42	84.15	85.63	86.55
Rank-T5 L2 loss (t=2)	82.46	71.98	85.34	91.54	94.86	97.14	71.98	80.40	83.51	84.94	85.82
Rank-T5 L2 loss (t=3)	81.67	70.51	85.12	90.94	94.86	97.61	70.51	79.73	82.64	84.33	85.39
Rank-T5 L2 loss (t=5)	83.70	73.67	86.86	92.41	95.59	97.82	73.67	81.99	84.77	86.14	87.00
Rank-T5 L3 loss (t=1)	**85.29**	**76.06**	**88.47**	**93.30**	**96.25**	**98.34**	**76.06**	**83.89**	**86.31**	**87.57**	**88.38**

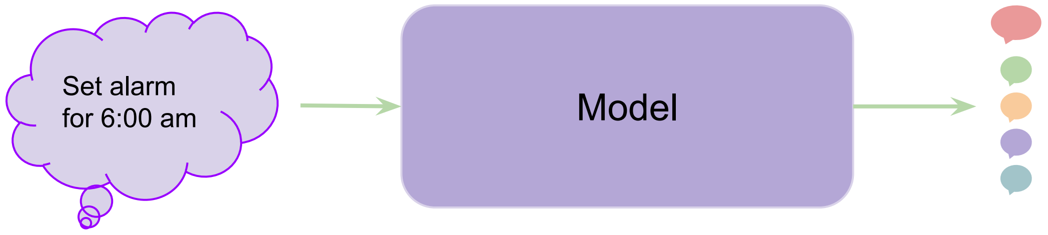
Results {Ablation Study} % of GMR in Train

	MRR	R@1	R@2	R@3	R@4	R@5	nDCG@1	nDCG@2	nDCG@3	nDCG@4	nDCG@5
Rank-T5 L2 loss (t=3) 50% GMR	**89.44**	82.45	**92.25**	95.95	**97.85**	98.91	82.45	**88.63**	90.48	**91.30**	**91.72**
Rank-T5 L2 loss (t=3) 0-50% GMR	**89.44**	**82.54**	91.94	**96.06**	97.82	**98.94**	**82.54**	88.47	**90.53**	91.29	**91.72**
Rank-T5 L2 loss (t=3) 100% GMR	88.85	81.80	91.32	94.94	97.44	98.78	81.80	87.81	89.62	90.69	91.21

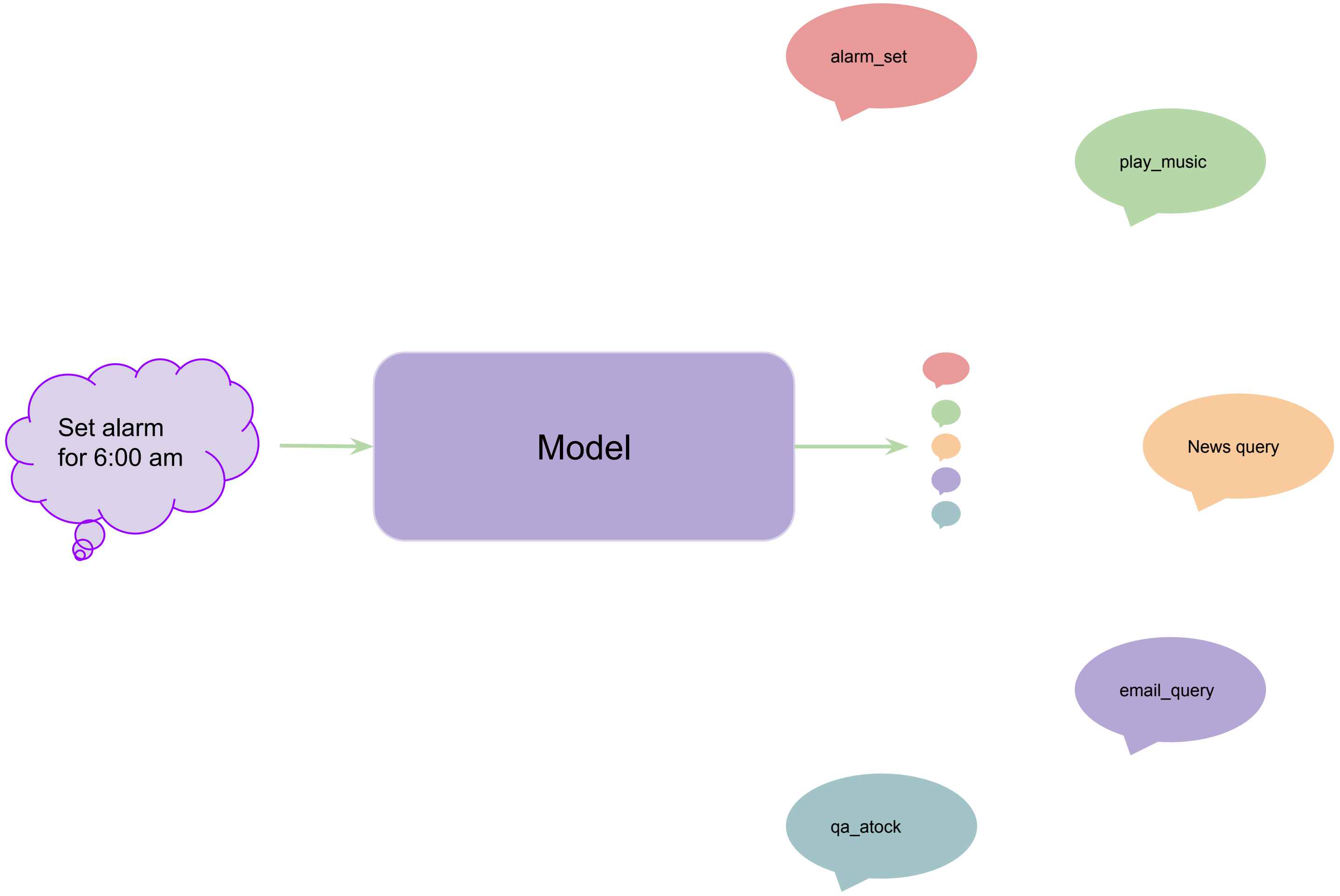
Intent Detection

Intent Detection

Intent Detection



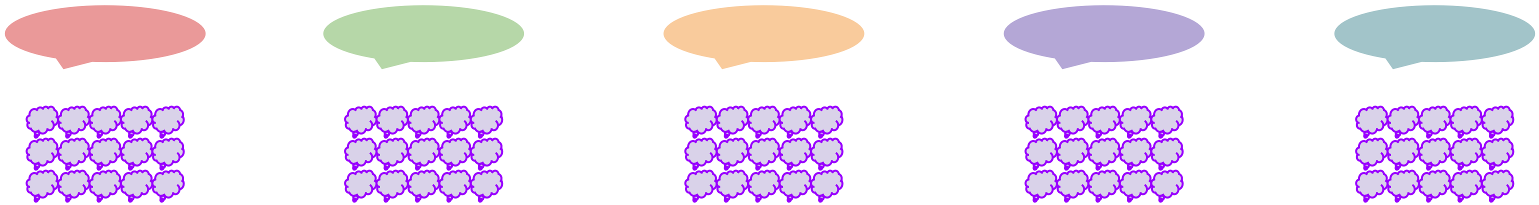
Intent Detection



Methodology

- Few Shot Intent Detection
- Datasets
- Contrastive Learning
- Curriculum Learning
- Domain Adaptation
- Results

Few Shot Intent Detection

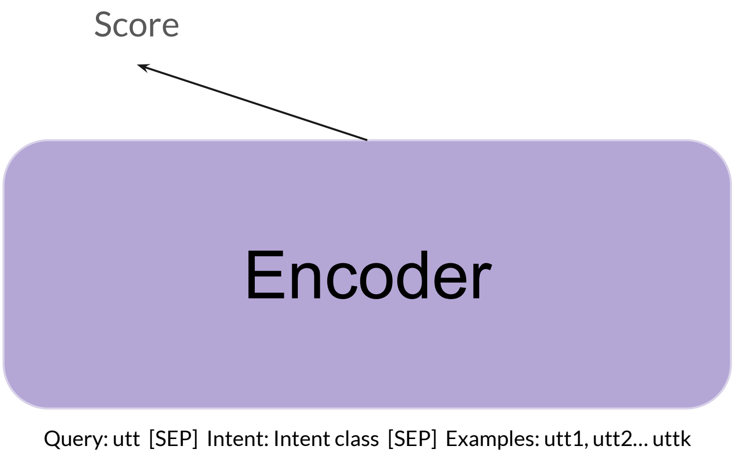


Massive Dataset and OOD Dataset

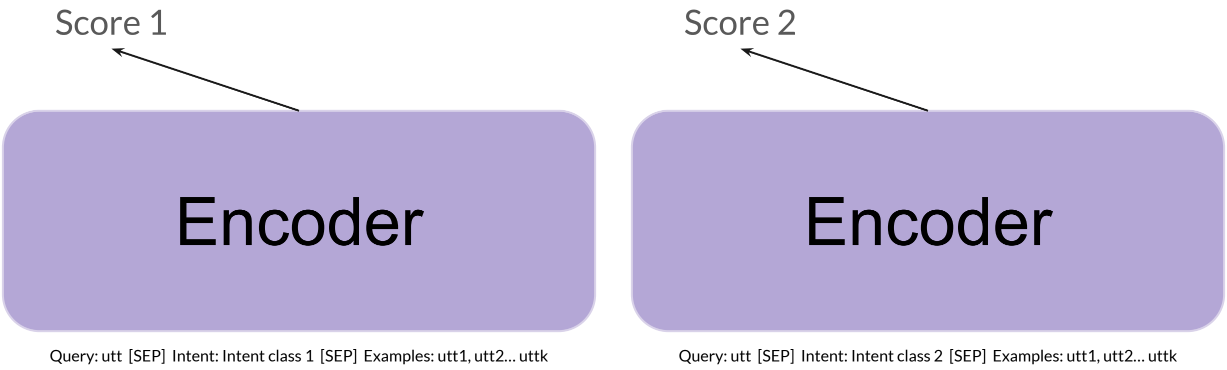
→ Massive has multiple languages, but focus is only on english
→ The dataset has around 11.5k examples in training split
→ It has 60 different intent types from 18 different scenarios.

→ OOD data set has 150 intent classes whose domains are quite different from massive's
→ Each intent class has only 15 utterances so total of 2250 examples in train split.
→ Test split has 6000 examples without the labels

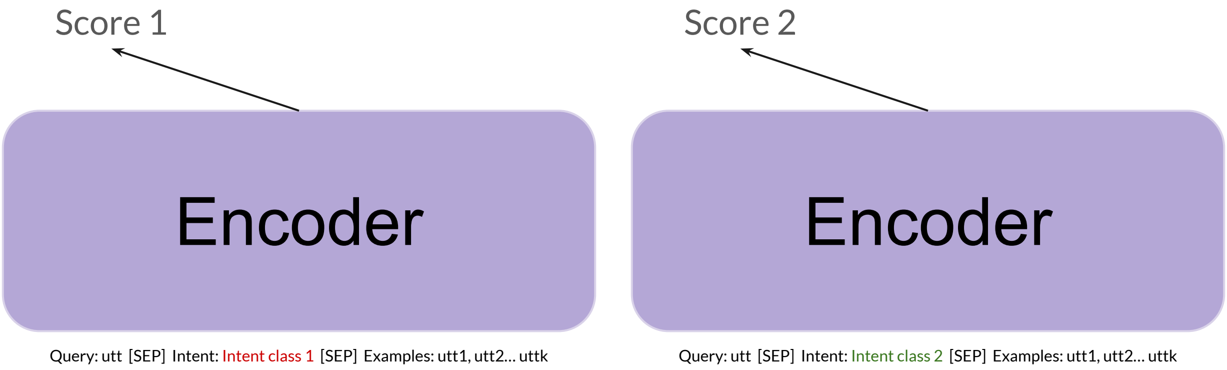
Model


Score
Encoder
Query: utt [SEP] Intent: Intent class [SEP] Examples: utt1, utt2… uttk

Contrastive Learning

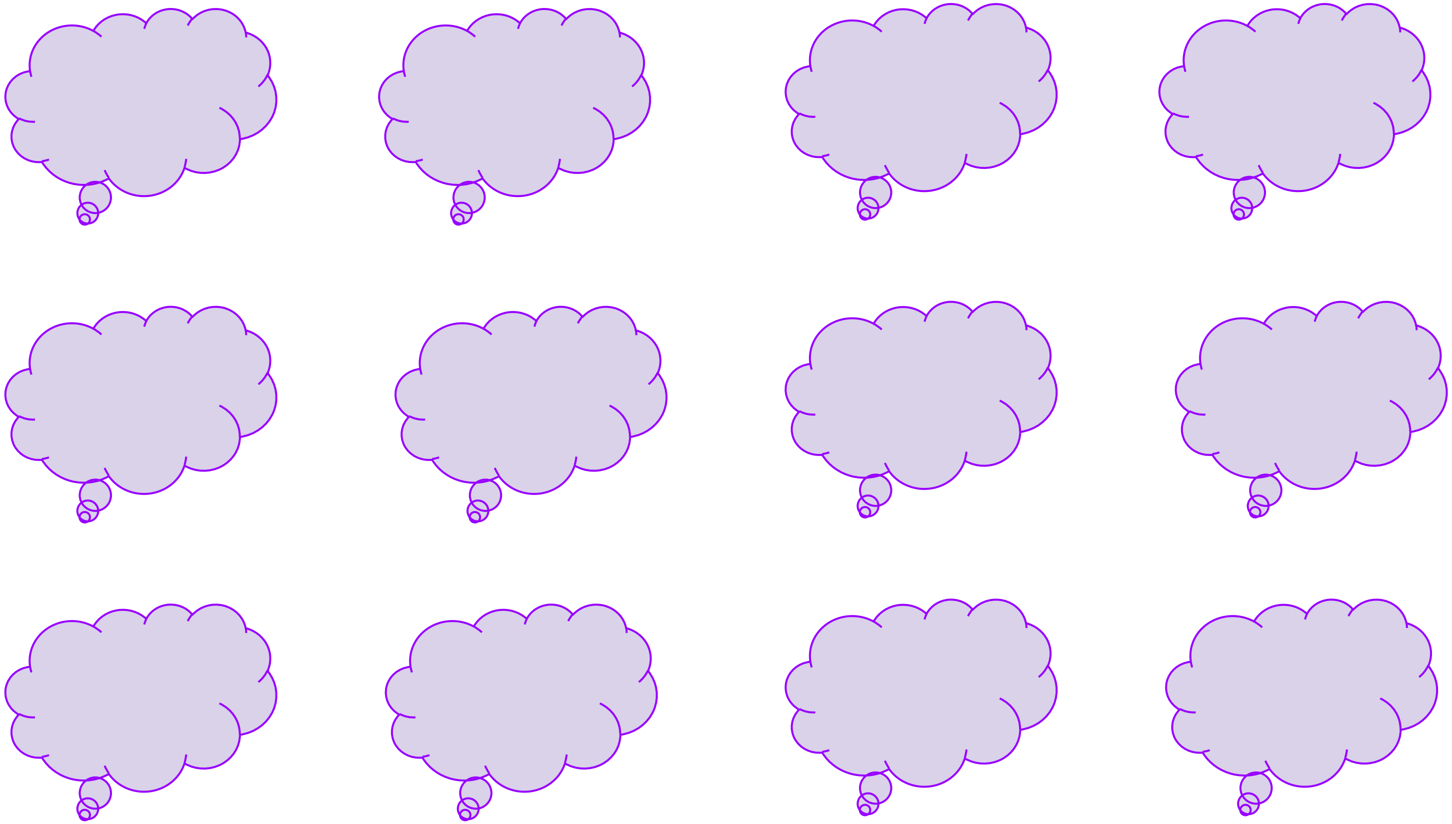


Contrastive Learning



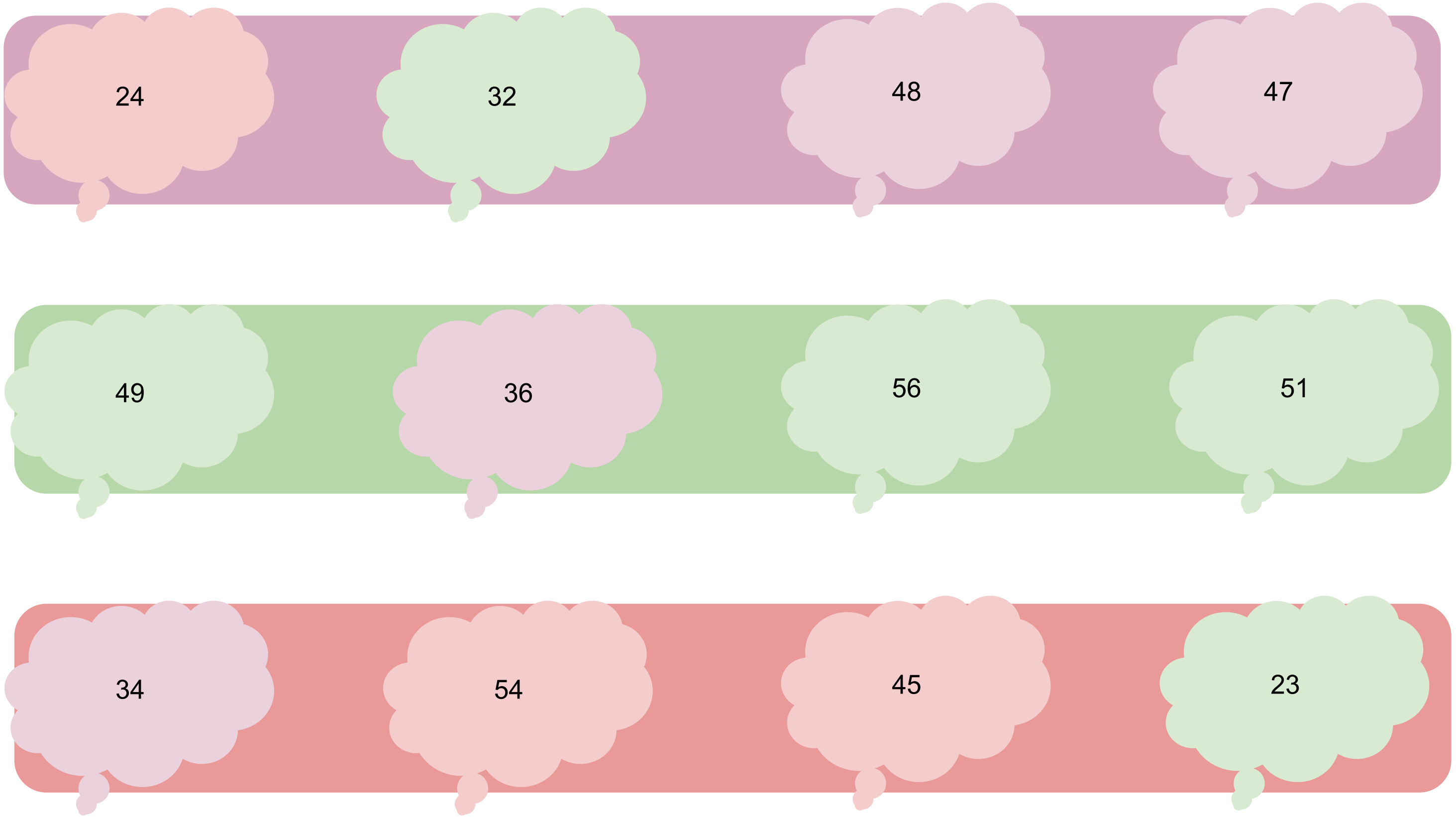
$$\text{Loss} = \frac{\text{Exp(Score 2)}}{\text{Exp(Score 1)} + \text{Exp(Score 2)}}$$

Curriculum Learning



Curriculum Learning

Curriculum Learning



Curriculum Learning

→ For a particular intent each intent is ranked based on the normalized cumulative prediction score, from high to low.
→ While data preparations, for selecting negative in initial steps we start with lowest scoring n intents and then gradually move towards the highest scoring n intents.

1	2	3
2	1	3
3	1	2

```
decay = [((k)**0.25)/((n**0.25)) for k in range(n2)]
decay2 = [((k)**0.75)/((n**0.75)) for k in range(n2)]
lspr1 = [int((1-x)*50) for x in decay2]
lspr2 = [int((1-x)*140) for x in decay]
conf = mass_conf[int2id[eg["intent"]]][id1:(id1 + 10)]
```

Curriculum Learning

→ For a particular intent each intent is ranked based on the normalized cumulative prediction score, from high to low.
→ While data preparations, for selecting negative in initial steps we start with lowest scoring n intents and then gradually move towards the highest scoring n intents.

1	2	3
2	1	3
3	1	2

```
decay = [((k)**0.25)/((n**0.25)) for k in range(n2)]
decay2 = [((k)**0.75)/((n**0.75)) for k in range(n2)]
lspr1 = [int((1-x)*50) for x in decay2]
lspr2 = [int((1-x)*140) for x in decay]
conf = mass_conf[int2id[eg["intent"]]][id1:(id1 + 10)]
```

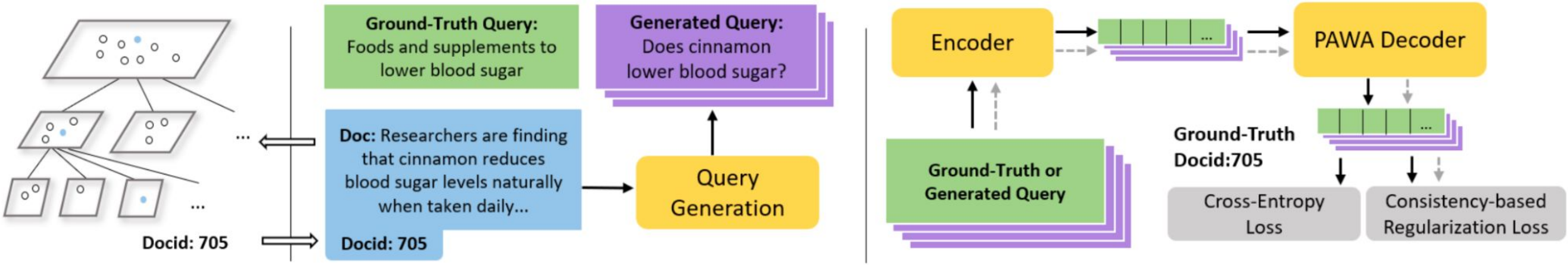
Results

	Accuracy	F1 Score	Precision
Train: Massive + OOD Unlabeled Test: OOD dataset W/O Contrastive Learning	74.52	73.93	79.13
Train: Massive + OOD Unlabeled Test: OOD dataset W/O Contrastive Learning	77.98	78.15	81.77
Train: Massive + OOD Unlabeled Test: OOD dataset W/ Contrastive Learning (random -ve)	81.87	81.78	82.98
Train: Massive + OOD Unlabeled Test: OOD dataset W/ Contrastive Learning W/ Curriculum Learning	84.60	84.49	85.68
Train: Massive + OOD Unlabeled Test: OOD dataset W/ Contrastive Learning W/ Curriculum Learning (Large Model FP16)	89.17	88.94	89.55

NCI-Contra

- Training Data + Objective
- Curriculum based Contrastive Learning

Neural Corpus Indexer: Training Data + Objective



(a) Preprocessing (b) Query Generation (c) Training pipeline of Neural Corpus Indexer

12-3-3 → (1,12)(2,3)(3,3)

1-12-3 → (1,1)(2,12)(3,3)

$$h_i = \text{TransformerDecoder}(x, h_1, h_2, ..., h_{i-1}; \theta_i),$$

$$p(r_i | x, r_1, r_2, ..., r_{i-1}, \theta_i) = \text{Softmax}(h_i W).$$

$$W_{ada}^i = \text{AdaptiveDecoder}(e; r_1, r_2, ..., r_{i-1}) W_i$$

Contrastive Learning

$$l_{\text{contra}}(y_i, g_i) = -\sum_{j=1}^{m} y_{ij} \log\left(\frac{e^{g_{ij}}}{\sum_{j'} e^{g_{ij'}}}\right)$$

$$g_{ij} = \prod_{k=0}^{n} P(d_{ij}^{k})$$

→ Documents are split in two sets, first set of documents contains their respective questions while training, questions related to documents in set two are only present in zero shot evaluation. But random 64 token of the documents are contained in the training.

Curriculum

```
prob = [(k**0.55)/((n**0.55)) for k in range(n2)]
```

→ Initially set the negatives for all the question as random passages but as the the training proceeds negatives are replaced with passages similar to the actual provenance

→ To identify similar passages, Used the baseline model to order passages based their respective scores as shown in the previous slide.

→ Then took the the mean by accumulating all the questions that belongs to a particular passage, and took the passages with with the highest scores as the similar passages.

→ All these are done in preprocessing and this is done for 5 epochs.

Results(Normal Evaluation, NQ)

	R@1	R@10	R@100	MRR@100
Without Contrastive Loss	56.34	78.93	86.54	69.78
With Contrastive Loss	56.51	79.18	86.62	68.99
With Curriculum Learning	**58.13**	**80.84**	**88.04**	**71.17**

Results(Zero-Shot Evaluation, NQ)

	R@1	R@10	R@100	MRR@100
Without Contrastive Loss	44.21	69.78	82.14	62.67
With Contrastive Loss	46.02	70.32	82.78	64.31
With Curriculum Learning	**48.37**	**72.15**	**83.81**	**66.00**

Conclusion and Future Work

Further Improvements

→ Scoring Text
→ Knowledge Distillation
→ Unified Architecture

→ Curriculum Learning
→ Domain adaptation
→ Few Shot Classification
Using Autoregressive Model

→ Experiments

References

- ➔ KILT: a Benchmark for Knowledge Intensive Language Tasks
- ➔ Wizard of Wikipedia: Knowledge-Powered Conversational agents
- ➔ Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering
- ➔ Distilling Knowledge from Reader to Retriever for Question Answering
- ➔ Query Enhanced Knowledge-Intensive Conversation via Unsupervised Joint Modeling
- ➔ Open-Domain Question Answering Goes Conversational via Question Rewriting
- ➔ RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses