INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE - INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE

Does Structure Matter? Leveraging Data-to-Text Generation for Answering Complex Information Needs

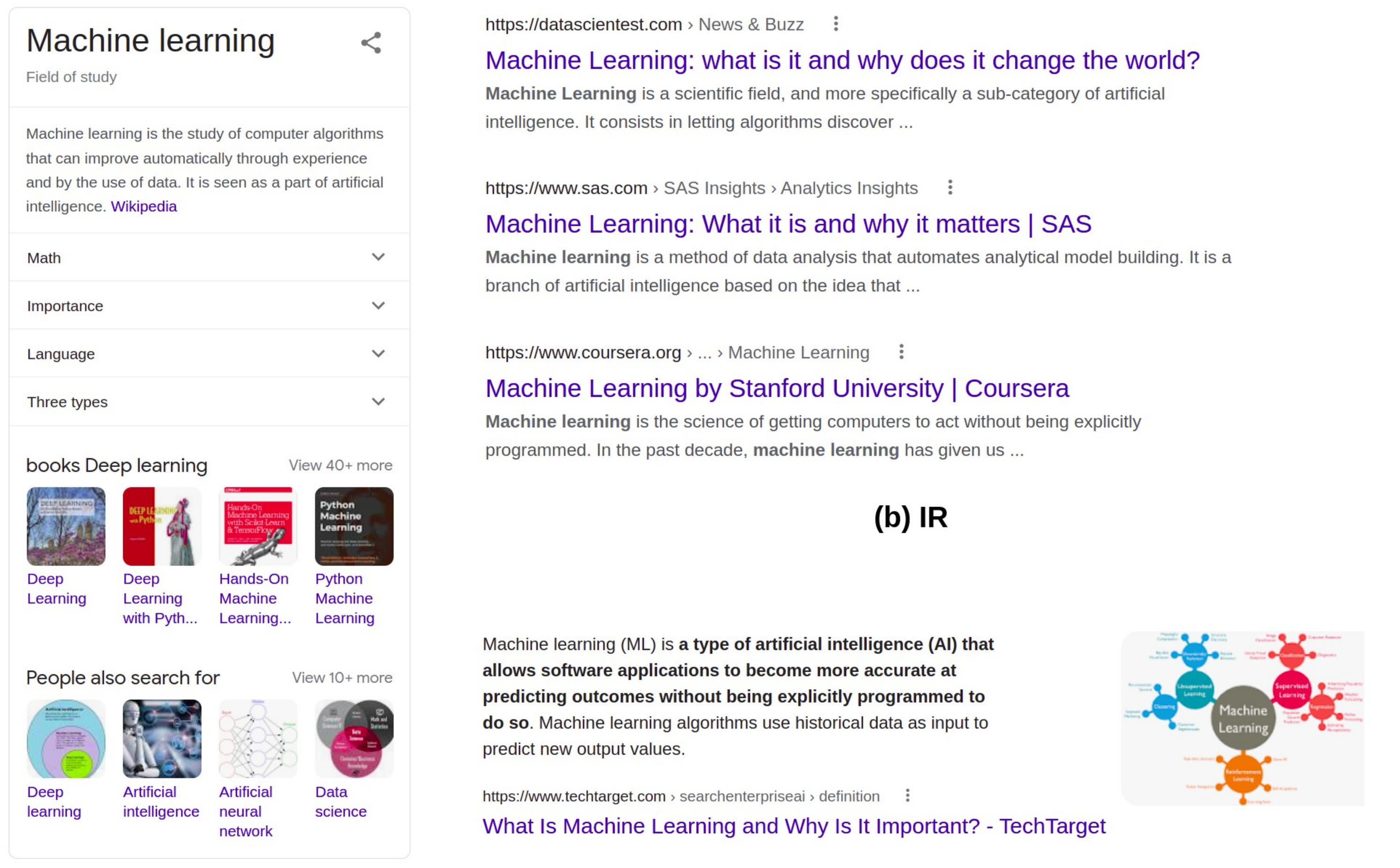
Hanane Djeddal, Thomas Gerald, Laure Soulier. Karen Pinel-Sauvagnat, Lynda Tamine

Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique (ISIR), in collaboration with Institut de Recherche en Informatique de Toulouse (IRIT)

Team Machine Learning and Information Access

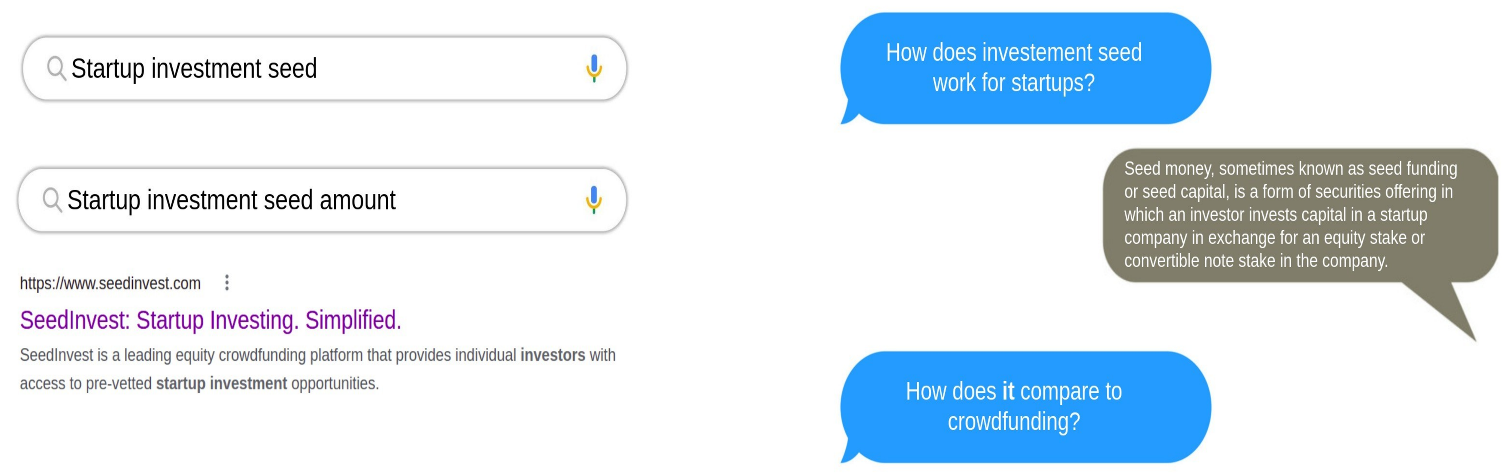
Context, Motivation and Objectives

- Contrary to simple information needs where a short and concise answer is sufficient; complex information needs require a detailed answer that tackles different aspects of the topic.
- Different solutions exist to present a search results:



(a) Aggregated IR (b) IR (c) QA

Conversational Search VS Traditional Search



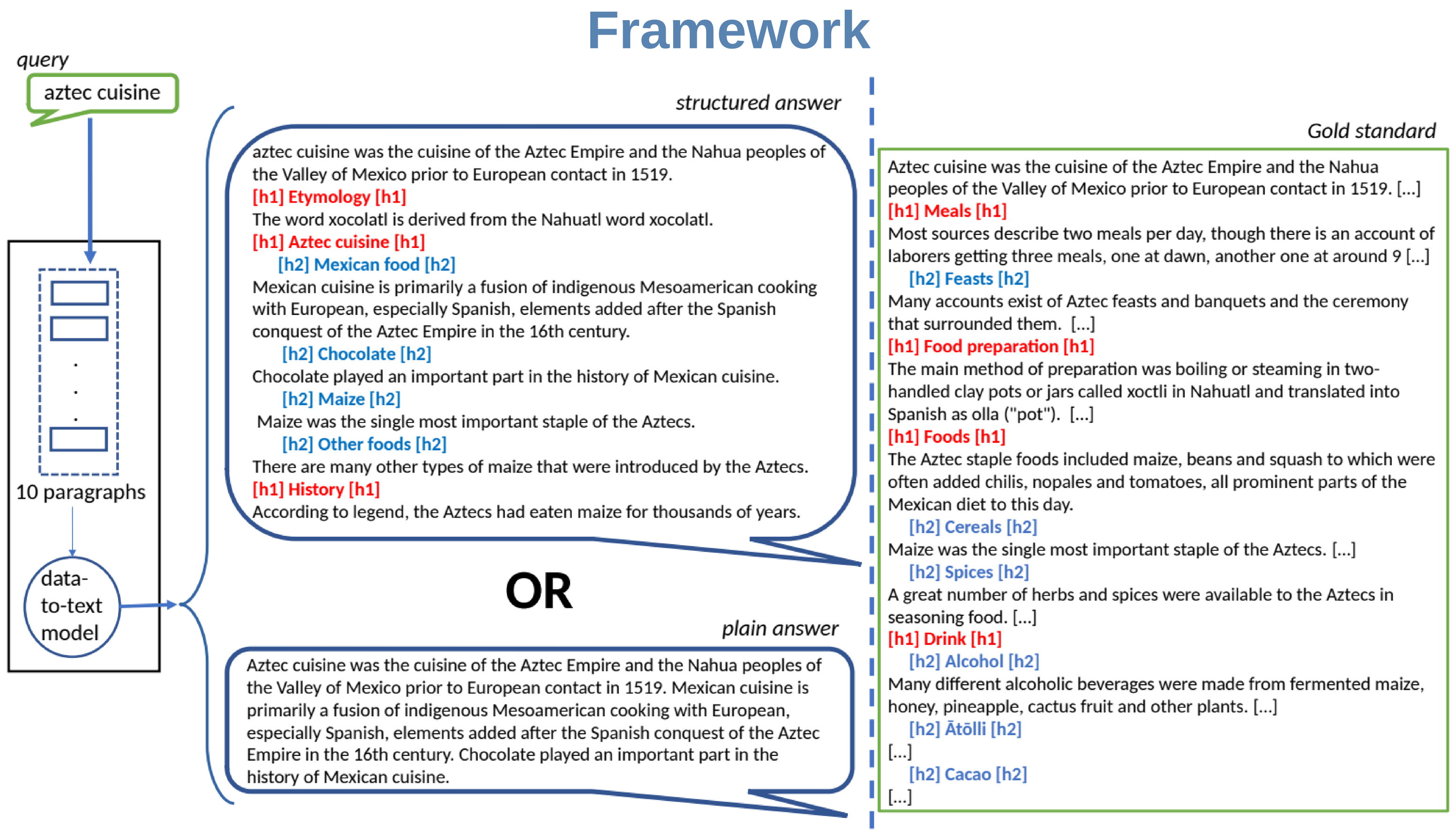
- Conversational search is more adapted for open-ended questions:
  - The user can express their needs in natural language.
  - Multi-turn dialogue.
  - Clarification questions.
  - Returning answers in natural language.

Objectives

- Provide a complete and **structured** answer in **natural language** that covers **the multiple facets** of an open-ended query, using as input an initial ranked list of documents.
- Solve complex information needs with generative models, particularly from the perspective of data-to-text generation.

A Data-to-Text Approach for Answer Generation

Framework



- Data-to-Text Generation introduces the notion of structure at two levels:
  - At input level (encoding): exploiting the structure of the inputs (cells, rows, etc)
  - At output level (decoding): structuring the output text by means of content selection and planning.

Content Selection and Planning Pipeline

- **Planning encoder-decoder**: Encodes each document $d_q \in D_q$ concatenated with the query q and decodes a plan p.

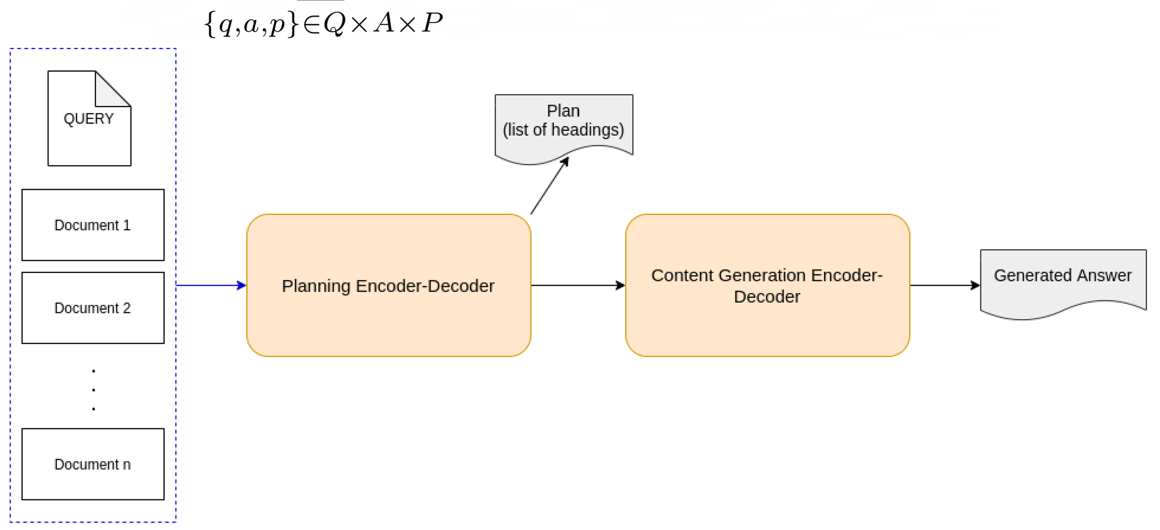
$$\mathcal{L}_{planning}(q,p) = P(p|q,\mathcal{D}_q) = \prod_{j=1}^{|p|}\prod_{k=1}^{|h_j|} P(h_{jk}|h_{j,<k},q,\mathcal{D}_q)$$

- **Content generation encoder-decoder**: Encodes each heading h_p in the plan p, concatenated with the embedding of the document list D_q and decodes an answer a.

$$\mathcal{L}_{answer}(q,a,p) = P(a|q,p,\mathcal{D}_q) = \prod_{k=1}^{|a|} P(a_k|a_{<k},q,p,\mathcal{D}_q)$$

- **Final Training Loss**:

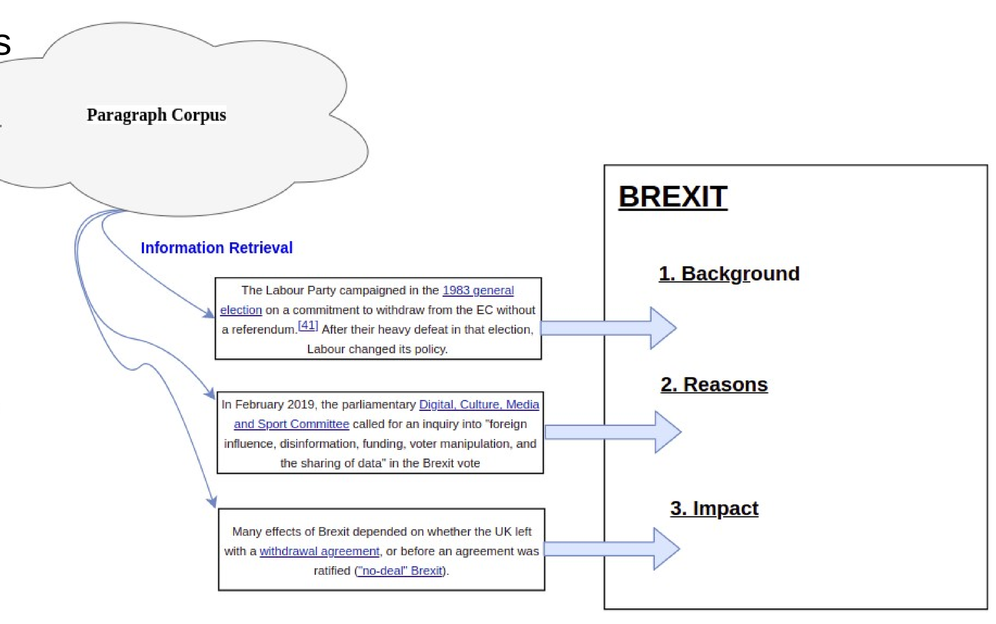
$$\mathcal{L} = \sum_{\{q,a,p\}\in Q\times A\times P} \mathcal{L}_{planning}(q,p) + \mathcal{L}_{answer}(q,a,p)$$



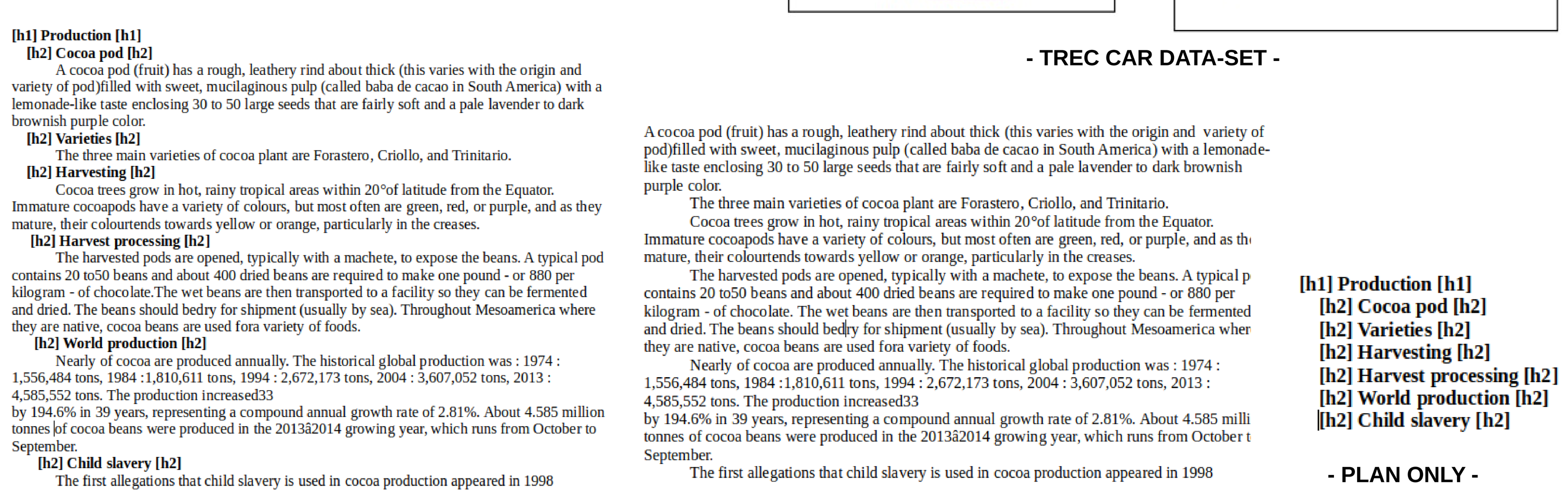
Evaluation Setup & Results

Data-Set

- Redefined the inputs and outputs of TREC CAR
  - Inputs : The query (the topic) and the top 10 most relevant passages (with BM25).
  - Output: Wikipedia article. Different variations were considered:
    - Plans only (Planning Module)
    - Structured or plain answer (Content Generation Module)



- TREC CAR DATA-SET -



- STRUCTURED ANSWER - - PLAIN ANSWER - - PLAN ONLY -

Results

		# tokens	Rouge-P	Rouge-R	Rouge-F	BERTScore	QuestEval
structured answers	EXT	898.22	36.50	26.99	29.86	85.50	41.99
	T5	126.25	**76.19**	08.41	14.25	**84.95**	39.06
	Planning-seq	181.39	62.94	09.57	15.36	84.44	37.47
	Planning-e2e	203.48	63.4	**10.21**	**16.09**	84.91	**39.31**
plain answers	EXT	885.35	34.35	26.73	28.99	86.30	42.34
	T5	110.62	**78.05**	09.24	15.48	85.51	39.89
	Planning-seq	163.58	65.73	10.34	16.27	84.29	38.46
	Planning-e2e	126.91	75.92	**10.34**	**17.05**	**85.67**	**40.78**

- Results of the answer generation -

		#tokens	#heading	depth	Rouge-P	Rouge-R	Rouge-F	BERTScore	Meteor
T5	FP	1.41	2.24	1.14	**39.89**	04.69	07.69	77.40	3.24
Planning-seq	IP	1.83	**4.42**	**1.45**	31.20	**8.29**	**11.51**	**81.25**	**5.97**
	FP	**1.88**	4.11	**1.45**	31.31	7.93	11.03	80.49	5.55
Planning-e2e	IP	1.57	3.37	1.15	35.15	07.34	11.12	81.21	5.51
	FP	1.64	3.27	1.16	34.79	06.38	09.78	80.70	4.71

- Analysis of the intermediate and final plans -

- Planning models produce longer text (200tokens), and do well in semantic-evaluation metrics.
- On the plain answers setting, our models are the most effective, which confirms the importance of structure prior even if it's not explicitly present in the final output.
- Our plans cover more facets, in correct order with a better relevant semantics.

Conclusion

- Generating a complex answer in a single turn to open-ended queries proves to be challenging because of the length of both inputs and outputs.
- Modeling a structure prior is beneficial to guiding the final output generation.
- Data-to-Text generation approaches provide a promising framework that can be applied to this task.

{hanane.djeddal,thomas.gerald,laure.soulier}@isir.upmc.fr
{sauvagnat,Lynda.Lechani}@irit.fr

Under joint supervision : Funded by: agence nationale de la recherche