Literature Review

Introduction

The main aim of this study is to find the possible alternatives of generating study materials and the best way of doing it. The problem is whether the current architecture has a unified and efficient method or should we progress with a “single model - single task” objective. Questions can be in various formats such as yes/no or true/false questions, multiple choice question or kind of wh-questions types, fill in the blanks and match the following or keyword extractions problems. Moreover, flashcards are also necessary for such a system.There has been lots of work targeting the question generation using encoder-decoder architectures. Encoder only architectures such as BERT (Devlin et al., 2019) due to its limited ability and are designed to single prediction per token or single prediction for entire output sequences. Encoder architectures like these are mostly used for rather simple tasks and not optimal for text generation tasks. However, they work best in situations where the primary objective is to learn the perfect language representation. On the other hand, text generation with auto-regressive or causal language models such as GPT-2 (Radford et al., 2019) allows us to train them on large amounts of unlabeled data and generate output more quickly. They are very reliable when we want zero-shot or few-shots inference but require a huge dataset and architecture to start with. On the other hand, encoder-decoder architecture provides us with a more task specific and light-weight solution but is more task specific. These trade-offs between architectures and understanding them are more complex than the problem itself. In this review, we delve into the different areas related to the problem.

General Review

Major Ideas

Each sub-problem requires a different approach to solve.

- For generating MCQs we require the model to generate questions and/or answers given the context and/or answer. Distractors can then be generated from answers through word sense disambiguation.

- Fill in the blanks and Match the following only requires keyword/keyphrase extraction strategies.

- For True/False or Yes/No questions there are several options. We can either use extractive or abstractive summarization techniques to generate True questions and we can gather the False questions by changing the noun phrases or verb phrases, changing the named entity, changing adjective, changing the main verb etc. To identify the verb phrases or noun phrases or any other part of speech tags we can use constituency parsing. A language model can then be used for sentence completion to add or remove negation to the original question.

- Flashcards can be simply generated by using large language models without any training or fine-tuning on specific sets of data.

Datasets available

Datasets for fine tuning the architecture on downstream tasks such as question generation, sentence completion and constituency parsing are available widely. Some of them are briefly described below.

1. SQuAD 1.1 (Rajpurkar et al., 2016), the previous version of SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles.

2. SQuAD2.0 (Rajpurkar et al., 2018) combines 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on this set, systems must also determine when no answer is supported by the paragraph and abstain from answering.

3. Penn Treebank (Marcus et al., 1993), the English PTB corpus, and in particular the section of the corpus corresponding to the articles of Wall Street Journal, is one of the most known and used corpus for the evaluation of models for sequence labeling. The task consists of annotating each word with its Part-of-Speech tag. Generally, the training set contains 38, 219 and 912 344 tokens , the validation set contains 5527 sentences and 131768 tokens and the test set contains 5462 sentences and 129 654 tokens.

4. BoolQ (Clark et al., 2019) is a question answering dataset for yes/no questions containing 15942 examples. These questions are naturally occurring and each sample contains a triplet of (question, context, answer).

5. OpenBookQA (Mihaylov et al., 2018), is a new kind of question-answering dataset modeled after open book exams for assessing human understanding of a subject. It consists of 5,957 multiple-choice elementary-level science questions, which probe the understanding of a small “book” of 1,326 core science facts and the application of these facts to novel situations. It also includes a core science fact.

Task	Dataset Variant	Model	Exact Match	F1	Accuracy
Question Answering	SQuAD1.1 dev	T5-11B	90.06	95.64	-
Question Answering	squad_v2	RoBERTa Large	85.168	88.349	-
Question Answering	SQuAD2.0 dev	XLNet	87.9	90.6	-
Question Answering	BoolQ	Gemma 7B	99.419	-	99.419
Question Answering	BoolQ	T5-XXL 11B(fine-tuned)	-	-	91.2
Question Answering	OBQA	FLAN 137B(zero-shot)			78.4
Question Answering	OBQA	LLaMA 65B(zero-shot)	-	-	60.2
Word Sense Disambiguatio n	SensEval 2 Lexical Sample	kNN-BERT	-	76.52	-
Word Sense Disambiguatio n	SensEval 3 Lexical Sample	kNN-BERT	-	80.12	-
Constituency Parsing	Penn Treebank	Self-attentive encoder + ELMo	-	95.13	-
Constituency Parsing	CTB5	Kitaev etal. 2018	-	87.43	-

Table 1: Common benchmarks

Common models used for the problem

Various architectures that can be used as a possible solution to the problem are briefly discussed below.

1. T5: T5 (Raffel et al., 2023) is a encoder-decoder architecture model pre-trained on a multitask mixture of unsupervised and supervised tasks and for which each task is

converted into a text-to-text format. T5 works well on a variety of tasks out of the box by prepending a different prefix to the input corresponding to each task. The pretraining includes both supervised and self-supervised training. The T5 11B is a checkpoint with 11 billion parameters.

2. GPT-2: OpenAI GPT-2 is a causal (unidirectional) transformer pre-trained using language modeling on a very large corpus of ~40 GB of text data. Hence, it is powerful at predicting the next token in the sequence. This allows it to generate syntactically coherent text.

3. BERT: Bidirectional Transformer models like BERT, DistilBERT (Sanh et al., 2020), and RoBERTa (Liu et al., 2019) are pre-trained with language modeling and next sentence prediction on large corpora like the Toronto Book Corpus and Wikipedia. DistilBERT is a faster, smaller version of BERT with similar performance, while RoBERTa modifies key hyperparameters for better training efficiency.

4. Gemma : Gemma (Gemma Team et al., 2024), a new family of open language models demonstrating string performance across academic benchmarks for language understanding, reasoning and safety. The two variants are different in sizes (2B and 7B parameters), and provide both pre-trained and fine-tuned checkpoints. It is shared publicly. It is designed to perform various NLP tasks such as text generation, completion, translation and summarization.

5. FLAN 137B : It takes a 137B parameter pretrained language model. Thereafter, the model is Instruction tuned on over 60 NLP tasks verbalized via natural language instruction templates. This instruction-tuned model is FLAN (Wei et al., 2022). Evaluation of FLAN on unseen tasks surpasses zero-shot 175B GPT-3 on 20 of 25 tasks, even outperforms few-shot GPT-3 by a large margin on ANLI, BoolQ, OpenBookQA etc.

6. Self-attentive encoder + ELMo: Self-attentive encoder (Vaswani et al., 2023) is part of transformer architecture with an attention mechanism. ELMO (Peters et al., 2018), deep contextualized word in vectors, is beneficial for achieving top results in NLP tasks. It uses a bidirectional language model pre-trained on large text corpus, and when combined, can create a strong parser.

Major Issues with problem/models/datasets

The major issues with the above datasets and/or models are discussed below:

1. There are multiple techniques to approach the problem including the one that is described above. Each task can be independently solved using different techniques, and to find a good method is an iterative process and requires lots of knowledge.

2. Some LLMs such as GPT-2 are not open language models and are very large in size with approximately 1.5B parameters. Likewise, FLAN 137B has 137B parameters. The computational resources they require are huge. As an example, the instruction tuning takes around 6-hours on a TPUv3 with 128 cores. Lastly, there are family of models to choose and finding a balance between these is itself a research question.

3. Though datasets are available for related tasks, they are not specific to the problem and we must adapt it to our problem. This might include creating custom metrics, lack of standard benchmarks etc. Likewise, every dataset found online that is relevant to our problem is imbalanced. For example, SQuAD1.1 has 100,000+ samples and BoolQ has 15942 samples and each solves different tasks. Lastly, the OpenBookQA dataset requires the system to apply facts before finding an answer.

Supporting Articles and Documents

1. The Stanford Question Answering Dataset - https://rajpurkar.github.io/SQuAD-explorer/
2. A python implementation of the parsers described in *"Constituency Parsing with a Self-Attentive Encoder - "*https://spacy.io/universe/project/self-attentive-parser
3. Constituency Parsing - https://web.stanford.edu/~jurafsky/slp3/old_sep21/13.pdf
4. Exploring Variants of BERT - https://www.scaler.com/topics/nlp/bert-variants
5. Byte-Pair Encoding tokenization https://huggingface.co/learn/nlp-course/en/chapter6/5
6. Penn Treebank Dataset - https://paperswithcode.com/dataset/penn-treebank
7. NLTK:: Sample usage for wordnet - https://www.nltk.org/howto/wordnet.html
8. Gemma Family of Models - https://www.kaggle.com/models/google/gemma

Paper Review

1. Using BERT For Word Sense Disambiguation

Title: Using BERT For Word Sense Disambiguation

Author: Du, J., Qi, F., & Sun, M.

Publication Date: 2019-09-18

Source Link: http://arxiv.org/abs/1909.08358

BERT improves polyseme representations for Word Sense Disambiguation (WSD), achieving state-of-the-art results on English All-word WSD evaluation.

The evaluation framework provides five all-words fine-grained WSD datasets for evaluation: Senseval-2 (Edmonds & Cotton, 2001), Senseval-3 task1 , SemEval-07 task 17, SemEval-13 task 12, SemEval-15 task 13. Training is done on Semcor and OMSTI.

BERT transformer encoder has 12 layers, hidden size is 768 and 12 attention heads.

Two encoder models are utilized in a system where one serves as a polyseme feature extractor while the other provides sense definitions. Outputs from both models are used by a two layer perceptron. The optimizer is adam. Learning rate decreases during training, starting at 0.001 and decreasing by i in each epoch. BERT encoder parameters remain fixed in the first 10 epochs, with training continuing for 50 epochs to select the model with the highest F1-score on the validation set.

To address the classifier poor performance on unseen or infrequent words the paper introduces the use of sense definition to address the issue of data scarcity.

BERTdef and BERT models, with/without sense definitions, outperform Lesk+ext,emb, Babelfy, Bi-LSTM on Senseval-2, Senseval-3 task1, SemEval-13 task12, SemEval-15 task 13 datasets.BERTdef shows 8% F1-score improvement on unseen words over BERT.

Supervised methods perform better than knowledge-based methods, but knowledge based methods are usually unsupervised and require no sense annotated data.

2. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Author: Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.

Publication Date: 2023-09-19

Source Link: http://arxiv.org/abs/1910.10683

Introducing a unified framework that converts all text-based language problems into a text-to-text format.

The Colossal Clean Crawled Corpus (C4) dataset underwent cleaning heuristics to process text extracted from Common Crawl. Common Crawl is a web archive offering non-markup text by eliminating irrelevant content. The resulting dataset is substantial, mostly free from unwanted text like offensive language, placeholders, and code.

“Text-to-Text Transfer Transformer” **T5** is an encoder-decoder Transformer. The paper itself is about this Transformer. The code base link for the transformer architecture is provided below.
https://github.com/google-research/text-to-text-transfer-transformer

Pre-training models in an unsupervised manner imparts general knowledge for future tasks. It uses "masked language modeling," where the model predicts missing tokens, inspired by BERT. The aim is to train a T5 model for various benchmarks like summarization, translation, and more. Tasks are converted into a "text-to-text" format by adding task-specific prefixes to input samples. For example, "translate English to German: That is good." will output "Das ist gut.". Despite being a hyperparameter, the prefix wording has minimal effect on performance.

The masking approach here used in the decoder part of the transformer is slightly different. When attending over a prefix the masking pattern is different. Instead of using a causal mask, it uses fully-visible masking during the prefix portion of the sequence and causal masking thereafter.

Various evaluation metrics are used for different tasks: SQuAD-Exact match, CNN/Daily Mail-ROUGE, WMT English to French and German-BLEU, GLUE and SuperGLUE-average scores for all subtasks. Scores for encoder-decoder with denoising objective: SQuAD(80.88), CNN/Daily Mail(19.24), WMT English to French(39.82), GLUE(83.28). Results vary based on architecture used, with encoder-decoder architecture with denoising objective excelling. Despite being twice the size of decoder-only models, it has the same computational cost. Sharing parameters across encoder-decoders yields similar performance. Performance drops with fewer layers. Using denoising objective improves downstream task performance compared to language modeling objective. Various unsupervised objectives like Prefix language modeling and BERT-style are available for general-purpose knowledge. BERT-style objective performs the best overall.

The pre-training provides significant gains across almost all benchmarks. The only exception is **WMT** English to French, which is a large enough data set that gains from pre-training tend to be marginal. The results suggest that additional exploration of objectives similar to the ones considered here may not lead to significant gains for the tasks and models considered here. Instead, it may be fortuitous to explore entirely different ways of leveraging unlabeled data.

3. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Title: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Authors: Devlin, J., Chang, M., Lee, K., Toutanova, K.

Publication Date: 2019-05-24

Source Link: http://arxiv.org/abs/1810.04805

BERT, short for Bidirectional Encoder Representations from Transformers, pre-trains deep bidirectional representations from text by considering both left and right context. It can be fine-tuned for various tasks without major modifications.

The pretraining was carried out on **BookCorpus**(800M words) and **English Wikipedia** (2,500M words). For Wikipedia only text passages were extracted and lists, tables and headers ignored.

BERT's model architecture is a multi-layer bidirectional Transformer encoder based on original implementation. Let L, H, A be the number of layers, hidden size and number of attention heads respectively. The **BERT-base** has L=12, H=768, A=12, Total Parameters=110M and BERT large has L=24, H=1024, A=16, Total Parameters=340M. The code base link is provided below.
https://github.com/google-research/bert

The paper outlines two steps: pre-training and fine-tuning. During pre-training, BERT learns from unlabeled data through masked language modeling and next sentence prediction. WordPiece (Wu et al., 2016) Tokenization is applied for BERT. For next sentence prediction, 50% of the time the following sentence is the actual one, and the other 50% it is random. The [CLS] token output is crucial in determining the coherence between two sentences. Overall, pre-training involves training the model on diverse tasks using unlabeled data before fine-tuning for specific applications.

The paper discusses different architectures like GPT and ELMo for processing text. GPT uses a left-to-right transformer, ELMo uses two independently trained models concatenated together. BERT, on the other hand, uses a bidirectional approach to compute hidden states on both right and left contexts simultaneously. This allows for a more effective text analysis.

GLUE (Wang et al., 2019**)** Benchmark consists of a number of tasks such as MNLI, QQP, QNLI, SST-2, CoLA, STS-B, MRPC and RTE. F1 scores are used for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores for the rest of the tasks. Average score across all these tasks are reported. Average Scores: Pre-OpenAI SOTA (74.0), BiLSTM+ELMo+Attn (71.0), OpenAIGPT (75.1), BERT-base (79.6) and BERT-large (82.1). Both BERT-base and BERT-large outperforms GPT and ELMo on all tasks by substantial margin.

Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems.

4. Language Models are Unsupervised Multitask Learners

Title: Language Models are Unsupervised Multitask Learners

Author: Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.

Source Link: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

The paper highlights that fine tuning on downstream tasks i.e. the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. The paper claims that the **GPT-2** model the transformer achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting.

This model stands out from previous ones by training on a wide range of text sources, rather than just one domain like news or fiction. The goal is to capture diverse natural language examples from different contexts. While Common Crawl provides a vast data source, the quality may be lacking. To improve this, pages curated by humans are used. To streamline the process, outbound links from popular Reddit posts were gathered. The resulting WebText dataset comprises text from 45 million links, with additional processing like extracting text from HTML and excluding Wikipedia pages.

A Transformer based architecture, largely follows the details of the **OpenAI GPT** model with a few modifications. The vocabulary has expanded to 50,257. The context size is increased from 512 to 1024 tokens and batch size of 512. The code base link is provided below.
https://github.com/openai/gpt-2

Language modeling involves estimating distributions without supervision using variable length sequences. Systems like GPT-2 can handle various tasks by conditioning not only on input but also on the task to be performed. Rather than just probability(output|input), it models probability(output|input, task). GPT-2 allows flexibility in specifying tasks, inputs, and outputs as symbols. For example, a translation task can be represented as (translate to French, English text, French text). Language modeling can learn tasks without explicit supervision on predicted outputs. Experiments show large language models can effectively perform multitask learning in this framework.

GPT-2 utilizes **Byte Pair Encoding** (Sennrich et al., 2016) tokenization with a base vocabulary size of 256 that expands during training as frequent pairs merge. However, BPE's greedy heuristic can lead to suboptimal merges, resulting in numerous variations like dog. dog! dog?. To address this, BPE restricts merging across character categories, except for spaces, enhancing compression efficiency with minimal word fragmentation.

Perplexity and Accuracy are mostly used to report the zero-shot results. GPT2, BERT, and original GPT were tested on datasets like LAMBADA, CBT-CN, WikiText2, enwik8. No training or fine-tuning was done. All models underfit WebText but perform well on various datasets, advancing the state of the art in a zero-shot scenario. Accuracy Scores of BERT: LAMBADA (45.99), CBT-CN (87.65), CBT-NE (83.4). Accuracy Scores of GPT-2: LAMBADA (63.24), CBT-CN (93.30), CBT-NE (89.05).

If a large architecture much like **GPT-2** which has a 1.5B parameter is trained on a large dataset including naturally occurring demonstrations of multiple tasks the model can perform downstream tasks in a zero-shot setting- without any parameter or architecture modification.

5. FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Title: FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Author: Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q. V.

Publication Date: 2022-02-08

DOI: 10.48550/arXiv.2109.01652

Source Link: http://arxiv.org/abs/2109.01652

The paper discusses a method to enhance zero-shot learning in language models using instruction tuning. Fine-tuning language models on datasets described with instructions boosts performance on new tasks.

The collection of web documents(including those with computer code), dialog data, and Wikipedia, tokenized into 2.49T **BPE** tokens with a 32k vocabulary using the **SentencePiece** library. Around 10% of the training data was non-English.Source code for loading the instruction tuning dataset used for **FLAN** is publicly available at https://github.com/google-research/flan. To save resources, existing datasets are converted into instructional formats for instruction tuning.

There are 62 datasets available on Tensorflow, covering language understanding and generation tasks. Each dataset is placed into one of twelve task clusters based on task type. For each dataset, ten templates are created, including variations to diversify instructions. The instruction tuning process begins using randomly selected templates from the mixture of datasets.

LaMDA-PT, a dense left-to-right, decoder-only transformer language model of 137B parameters is used. This model is pretrained on the dataset explained above. The **LaMDA-PT** only has language model pretraining (c.f. **LaMDA**, which was fine-tuned for dialog). **FLAN** is an instruction-tuned version of **LaMDA-PT**. **FLAN** (Finetuned Language Net) giant architecture of 137B parameters is pre-trained and subjected to instruction tuning on over 60 NLP datasets verbalized via natural language instruction templates.

FLAN architecture, unlike BERT or GPT-3, is pre-trained and fine-tuned on various tasks using natural language instructions. It undergoes zero-shot evaluation on new tasks after being trained on a mix of datasets with balanced sizes. The training examples are limited to 30K per dataset and a mixing rate of maximum 3k is applied. Models are fine-tuned for 30K gradient steps with a batch size of 8,192 tokens and adafactor optimizer with a learning rate of 3e-5. The packing procedure combines multiple training examples into one sequence, with inputs and targets separated by a special EOS token.

The model was tested on various tasks including natural language inference, reading comprehension, closed-book QA, translation, commonsense reasoning, coreference resolution, and struct-to-text. For each dataset, the mean of performance on all templates is used as a metric. Performance of FLAN is the mean of up to 10 instruction templates per task. While some task clusters showed strong results, instruction tuning did not benefit performance in tasks like commonsense reasoning or coreference resolution. FLAN outperformed LAMDA-PT on only three out of seven tasks, indicating limited usefulness of instruction tuning in language modeling tasks.

This paper has explored a simple method for improving the ability of language models at scale to perform zero-shot tasks purely based on instructions. This model compares favorably against GPT-3 and signals the potential ability for language models at scale to follow instructions.(Wei et al., 2022)

6. Constituency Parsing with a Self-Attentive Encoder

Title: Constituency Parsing with a Self-Attentive Encoder

Author: Kitaev, N., Klein, D.

Publication Date: 2018-05-02

Source Link: http://arxiv.org/abs/1805.01052

Self-attentive architecture improves information dissemination, achieving high accuracy on Penn Treebank and surpassing SPMRL language parsing. Neural network advancements enhance constituency parsing using encoder-decoder setups, with attention mechanisms effectively capturing global context.

The architecture features an encoder with self-attention layer and a specialized decoder for parsing - notably a chart parser with modifications. The highly performing parser serves as a solid starting point. The encoder is divided into two segments: one assigns context-aware representations to each sentence position, while the other combines word-based outputs to generate span scores. In addition to word embeddings, the encoder accepts part-of-speech tag embeddings from an external tagger for enhanced generalization. It also includes a table of position embeddings with consistent dimensions for all embeddings.

The model achieved a high score of 92.67 F1 on the Penn Treebank WSJ development set, while the model with the same decoder and LSTM-based encoder scored slightly lower at 92.24. By separating content and position information, parsing accuracy was improved, with query and key vectors being linear transformations of the word's position embedding. Disabling content-based attention in all 8 layers led to a small decrease in accuracy. The study also explored how combining content and position information in a single vector can affect attention balance. However, isolating position-related components did not significantly improve performance. Another approach, the factored scheme, showed promise with a development-set F1 score of 93.15.

The self-attention mechanism surpasses LSTM in performance. In the Content vs Position Attention Debate, position-based attention is crucial, while content-based attention is more beneficial in later layers. A study on using long-distance context information through windowing at test-time revealed that strict windowing leads to decreased accuracy, even a window of size 40. However, allowing the start token, first word, last word, and stop token to participate in all attention uses, while restricting pairs of other words within a given window, yielded slightly better outcomes. Removing tag embeddings caused a performance drop, mitigated by directly incorporating lexical features into the model. Strategies such as replacing words with frequently-occuring suffixes and using character-based Bi-LSTM in-place of part-of-speech tags improved parsing accuracy, with the use of external embeddings like ELMo further enhancing results (proved successful with development set result of 95.21 F1).

Therefore, the paper demonstrates that the choice of encoder can have a substantial effect on parser performance, in particular it reports state-of-the-art results with a novel encoder based on factored self-attention.

7. SENSEVAL-2: Overview

Title: SENSEVAL-2: Overview

Author: Edmonds, P., Cotton, S.

Publication Date: 2001-07

Source Link: https://aclanthology.org/S01-1001

This paper gives the overview of SENSEVAL-2: evaluation exercise, tasks, scoring system, results, and recommendations for future exercises. SENSEVAL, initiated in 1997 by ACL-SIGLEX, aims to evaluate WSD algorithms and systems' strengths and weaknesses across various languages and words, focusing on word sense disambiguation.

SENSEVAL-2 expanded to include new languages, evaluating WSD systems in three tasks across 12 languages: All-words (tagging most content words in a 5000-word text sample), lexical sample (tagging word samples in short text extracts), and translation-based sense distinction. Senseval task includes a lexicon with word-to-sense mappings, a manually tagged text corpus, and a sense hierarchy for refined scoring distinctions. WordNet was first used in Senseval for multiple languages.The gold standard corpus was divided into training and test sets for evaluation. Only 'unsupervised' systems could participate, unlike in English all-words tasks. The assessment on the test set is done with a 2:1 training to test ratio. All datasets are publicly accessible.

The University of Pennsylvania conducted an evaluation similar to the first SENSEVAL, with teams registering, downloading datasets, running systems on test data, submitting answers, and viewing results at a workshop. Tasks had specific timelines for data submission. The competition ran from April 17 to June 18, with deadlines for submissions.

Systems can assign multiple senses to a word with probabilities. Scoring requires at least one correct sense from the Gold Standard. Precision and recall evaluate systems, with recall measuring accuracy and precision measuring correctness. Coverage is also calculated. Systems are assessed for accurate and comprehensive sense identification within the task.

The attention changed from the winner to system performance. Scoring software bugs found, resulting in rescorings. Formatting errors allowed resubmissions. Panels covered domain-specific disambiguation, task design, sense distinctions, WordNets.

The workshop content should have been about the analysis of WSD algorithms, to gather information about systems (supervised, unsupervised, knowledge source etc.). The organization should be more democratic organization, become open and scientifically professional as possible.

Literature Review Matrix

Title/Author/ Date	Conceptual Framework	Research Question(s)/ Hypotheses	Datasets	Methodology	Analysis & Results	Conclusions	Implications for Future Research
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer /Raffel et al./2023	Leverage model's ability to maximize transfer learning efficiency and performance across diverse natural language processing tasks.	Introduce a unified system that can solve all the text based language problems to a text-to-text format.	Common Crawl archive is applied to many heuristics to obtain Colossal Clean Crawled Corpus (C4).	T5 model is pre-trained using unsupervised objective and study downstream performance on a diverse set of benchmarks .	Scores for encoder-decoder with denoising objective: SQuAD(80.88), CNN/Daily Mail(19.24), WMT English to French(39.82), GLUE(83.28)	Without much architectural modifications the model generalizes across multiple tasks.	Refining the model architecture, extending or fine-tuning the framework to user specific NLP tasks, investigating language-agnostic models.
Language Models are Unsupervised MultiTask Learners/ Radford et al./2019	Unsupervised multi-task learning with language models can enhance	Demonstrate that language models begin to learn NLP tasks	WebText dataset that includes natural language demonstrations in	Language modeling approach of training framed as an unsupervise	GPT-2 when compared with BERT and original GPT gave promising	A large architecture trained on a large dataset containing natural	One-shot or Few-shots learning can further enhance the model's adaptability

	their ability to generalize across diverse natural language processing tasks.	without any explicit supervision when trained on large dataset	varied domains and context as possible.	d setting where GPT-2 specify tasks, inputs and outputs all as a sequence of symbols.	results on 7 out of 8 datasets in zero-shot learning. Accuracy: LAMBADA (63.24), CBT-CN (93.30), CBT-NE (89.05).	language demonstrations can perform in a zero-shot setting.	on downstream tasks.
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding/ Devlin et al./2019	Leverage bidirectional contextual representation and large scale pre-training to enhance language understanding and performance.	Improve the fine-tuning based approach by using bidirectional representations of the encoder.	BookCorpus and English Wikipedia for training as well as task-specific datasets for evaluation.	Pre-training is carried out using MLM and NSP to improve the encoder's ability to learn representations.	Average Score on GLUE tasks: Pre-OpenAI SOTA (74.0), BiLSTM+ELMo+Attn (71.0), OpenAIGPT (75.1), BERT-base (79.6) and BERT-large (82.1)	Rich, unsupervised pre-training is an integral part of many language understanding systems.	Hints possible avenues for unsupervised objectives for pre-training, knowledge representation tasks (WSD and Keyword Extraction).
FINETUNED LANGUAGE MODELS ARE ZERO SHOT	Zero shot inference via natural language instruction templates	Instruction tuning the model and evaluation on several benchmarks	Collection of web documents, dialog data, computer code and	Pre-training the LaMDA-PT on language task and instruction-	Surpasses zero-shot 175B GPT-3 on 20 of 25 datasets and even	Instruction tuning models on collection of datasets improves	Gathering even more task clusters for fine-tuning, cross-

LEARNERS /Wei et al./ 2022	and key to its success.	that are unseen tasks i.e. zero shot learners.	some non-english data is present in the training set.	tune it on 60 NLP datasets, finally perform zero-shot inferences.	few-shot GPT-3 by a large margin on certain datasets.	zero shot performance.	lingual experiments ,using FLAN to generate data for downstream tasks.
Using BERT for Word Sense Disambiguation/ Du et al./2019	Contextual language understanding capabilities to enhance the accuracy and effectiveness of word sense disambiguation tasks through fine tuning on relevant datasets.	Propose BERT to extract better polyseme representations for WSD and explore several ways of combining BERT and the classifier.	Use SE7 as the validation set and SE2, SE3, SE13 and SE15 as test datasets, Trained on Semcor and OMSTI. Additional details in the paper review above.	Pre-trained BERT encoders are used for sentence and sense definition respectively. Outputs from these are passed to the softmax classifier to predict the correct sense.	BERT wins over Lesk, Babelfy, Bi-LSTM and many other systems.. BERTdef shows 8% F1-score improvement on unseen words over BERT.	Supervised methods outperform knowledge-based ones, yet knowledge-based methods need no annotated data.	Optimizing strategies, using RoBERTa for performance boost or DistilBERT to reduce size in exchange of accuracy or exploring hybrid models combing BERT to reach even higher.
Constituency Parsing with a Self-Attentive Encoder/ Kitaev & Klein/2018	The advent of neural networks, especially the ubiquity of transformer	Integration of self attention mechanisms into encoders can	The pretraining is done in the Penn Treebank WSJ or used pre-	Self-attention mechanism for encoder and chart parser with additional	Several techniques applied factor methods to find the importance	The choice of encoder can have a substantial effect on parser performanc	Further research into different ways can lead to additional improvemen

	s and its mechanisms, comparison of proposed architecture with the traditional models.	enhance accuracy and efficiency of constituency parsing models.	trained word representations. SPMRL dataset for benchmarking purposes.	improvements in the decoder side, for each tree generated we find the best one using score for each tree.	of context and position attention, windowed attention, external embeddings etc. (external embeddings development set result of 95.21 F1)	e, how different kinds of intricacies involved such as subword features or externally trained can have substantial effect.	ts in both parsing and other natural language processing tasks.
SENSEVAL-2: Overview/ Edmonds & Cotton/ 2001	Comprehensive evaluation of WSD systems discussing the exercise, the tasks, the scoring system and the results and future recommendations.	Effectiveness of the current WSD system in a standardized evaluation setting and solving the WSD-evaluation problem.	Lexicons such as Wordnet, a corpus of manually tagged text serving as the gold standard, and a sense hierarchy or grouping for refined sense distinctions in scoring.	Centrally managed system for delegation of the tasks and submission of the results following time constraints and scoring systems which used precision, recall and coverage metrics for evaluation.	Not a competition but focuses on knowing how each system performed. Bugs in the system were discovered, formatting errors made by some teams, allowed for resubmission.	Revealed the long-standing problem and also highlights the need for continued research and development in the field.	Majority of the workshop content should have been about the analysis of WSD algorithms, and gathering information about systems(supervised, unsupervised.

Reference: https://academicguides.waldenu.edu/ld.php?content_id=6154245

Project Solution Proposal

The advancements in Artificial Intelligence have achieved so much that the need for it has become mandatory in most of the field. Most importantly in the field of education, the need has increased due to the self-learner and remote learner. The importance of adaptive and personalized learning can be interesting for learning enthusiasts. Hence our project proposed a system called “learnX.ai (study support)”, an AI assisted study support system which generates MCQs, match the following, true/false questions, and flashcards. The features help the user by generating educational content with the use of NLP modes/models thus improving learning experiences.

QuestgenAI and Quillionz are AI systems available online for generating MCQs. Questgen offers one-click quiz generation, free plans, tutorials, customer support, and an API. Quillionz also provides free plans, customer support, an API, monthly free plans, and the ability to work with PDFs. Both systems aim to solve the same problem efficiently.

The study support system is planned to be made by combining tradittional as well as state-of-art NLP techniques. We aimed to design a system that can generate match the following, MCQs, flashcards, true/false questions, and fill-in-the-blanks with the help of techniques we selected from the research and findings. We selected models and their families like T5, Open AI GPT-2, Gemma, BERT etc.. We will mostly focus on the unified T5 model for generating all the contents due to its performance, size and computational efficiency. The BERT model or its variants are also proposed for generating true/false and fill-in-the-blanks. Some external frameworks are also already available for the task. GPT-2 or Gemma models can be used for generating flashcards. Even though these models can generalize well on other tasks as well on zero-shot inference, we hope to fine-tune the architecture for downstream tasks. We will also try to make use of the FLAN model for generating educational content and increase generalization capability in different question types.

The model requires a huge corpus of educational text from different domains to generate quality content for which we planned to collect the data from various datasets provided by the NLP community such as SQuAD, BoolQ, OpenBookQA etc. The data collected from the resources will be preprocessed to remove distractions, perform tokenization, and remove stop words and punctuation to retain quality content. We will use a training and fine-tuning approach in the preprocessed dataset. Thereafter, select the NLP model for training based on parameter efficiency, pre-training objectives, on-device computations, and improve zero-shot capability, and aim to fine-tune the pre-trained model on educational content. We will use the dataset to pre-train the selected models for a better understanding of the language and context. We will fine-tune the models with their contextual data for content generation with domain-specific data.

We intend to implement the FastAPI for processing the text and generating educational content at the user end. We aim to create a user interface using Streamlit for users to interact with the system.

The model's workflow is set up for the user to input educational text on the Streamit frontend and then click a button to generate the desired output. The input text is forwarded to the backend API using an HTTP post request, where it is processed and prepared for modeling analysis. The backend API uses NLP models to create educational content as per the user's request. The model produces and organizes content in a user-friendly manner, then sends it to the frontend for display to the user.

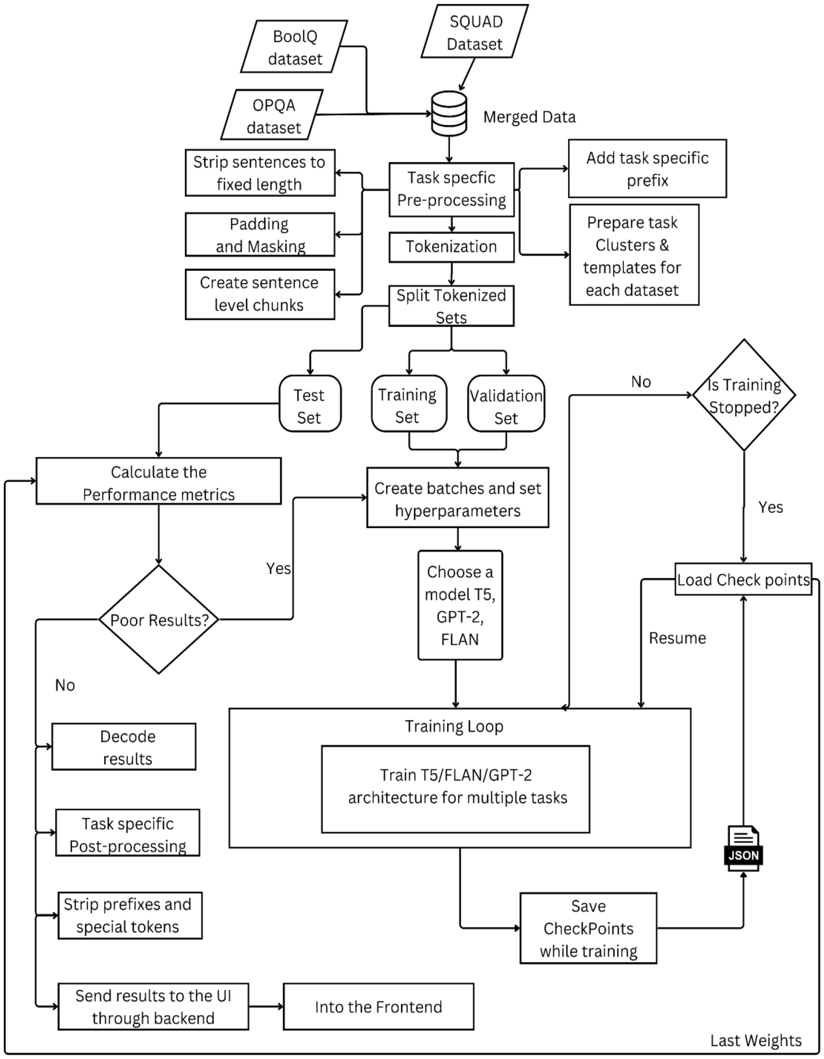
BoolQ dataset
SQUAD Dataset
OPQA dataset
Merged Data
Strip sentences to fixed length
Task specfic Pre-processing
Add task specific prefix
Padding and Masking
Tokenization
Prepare task Clusters & templates for each dataset
Create sentence level chunks
Split Tokenized Sets
Test Set
Training Set
Validation Set
No
Is Training Stopped?
Calculate the Performance metrics
Create batches and set hyperparameters
Yes
Yes
Choose a model T5, GPT-2, FLAN
Load Check points
Poor Results?
Resume
No
Training Loop
Decode results
Train T5/FLAN/GPT-2 architecture for multiple tasks
Task specific Post-processing
JSON
Strip prefixes and special tokens
Save CheckPoints while training
Send results to the UI through backend
Into the Frontend
Last Weights

Figure 1: System Diagram

1. The interface of the system is intended to be designed as a **Streamlit** application, where users interact by inputting the text and receiving the educational content they want to receive. The interface will include a text area and buttons for generating the content to match the following, flashcards, etc.

2. The system will integrate **FastAPI** Server as a bridge between the front end and the processing layers. It manages the HTTP requests and routes them to the specific processing modules.

3. Data pre-processing involves preparing the input text for model training or inference by processing it. The process involves eliminating punctuation and common words, splitting the text into individual words, and identifying key terms. This involves the use of the keyword extraction algorithm and text processing scripts. The tools utilized in this stage will involve the Huggingface API, NLTK, Spacy, and/or Gensim frameworks and libraries.

4. The NLP model intends to be used to generate educational content by transforming input text into fill-in-the-blanks, match-the-following, flashcards, and MCQ generation. Models like **T5**, **GPT-2**, **BERT**, **Gemma 7B**, **FLAN 137B** are supposed to be used. **HuggingFace** API offers various models for easy starting point.

5. The focus shifts to formatting the results of the NLP model into educational content using formatting scripts and response generation logic. Data post-processing involves various tasks to create the necessary content, such as clarifying word meanings, generating distractors, excluding certain tokens, and removing unnecessary prefixes.

6. The output generated is aimed to send back to the interface as a response to the input text for users. It can be managed with HTTP response handling and content display scripts.

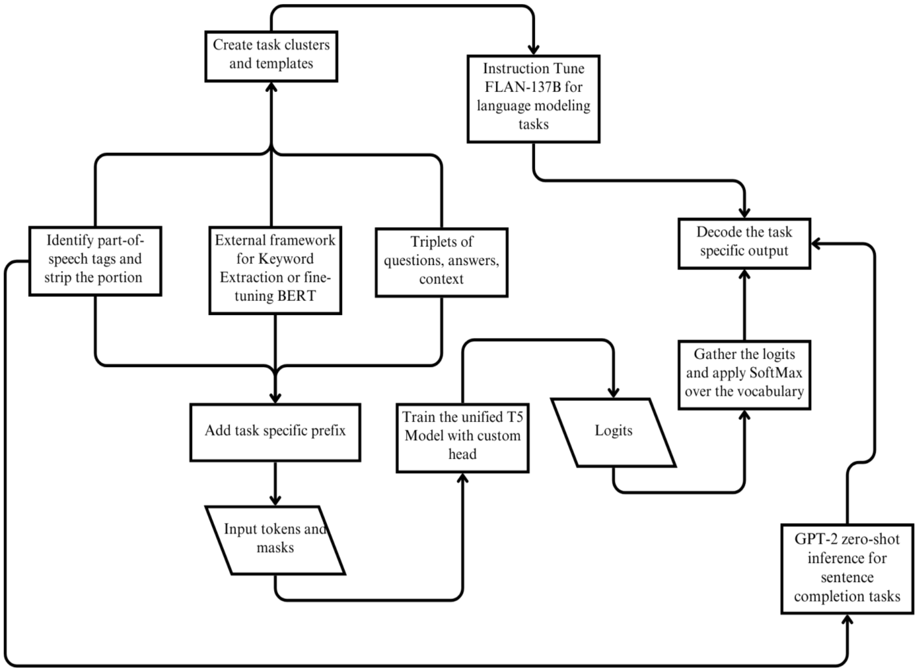
Create task clusters and templates
Instruction Tune FLAN-137B for language modeling tasks
Identify part-of-speech tags and strip the portion
External framework for Keyword Extraction or fine-tuning BERT
Triplets of questions, answers, context
Decode the task specific output
Gather the logits and apply SoftMax over the vocabulary
Add task specific prefix
Train the unified T5 Model with custom head
Logits
Input tokens and masks
GPT-2 zero-shot inference for sentence completion tasks

Figure 2: Block Diagram

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. http://arxiv.org/abs/1810.04805

Du, J., Qi, F., & Sun, M. (2019). *Using BERT for Word Sense Disambiguation* (arXiv:1909.08358). arXiv. http://arxiv.org/abs/1909.08358

Edmonds, P., & Cotton, S. (2001). SENSEVAL-2: Overview. In J. Preiss & D. Yarowsky (Eds.), *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems* (pp. 1–5). Association for Computational Linguistics. https://aclanthology.org/S01-1001

Kitaev, N., & Klein, D. (2018). *Constituency Parsing with a Self-Attentive Encoder* (arXiv:1805.01052). arXiv. http://arxiv.org/abs/1805.01052

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (arXiv:1910.10683). arXiv. http://arxiv.org/abs/1910.10683

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). *Finetuned Language Models Are Zero-Shot Learners* (arXiv:2109.01652; Version 5). arXiv. https://doi.org/10.48550/arXiv.2109.01652

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (arXiv:1907.11692). arXiv. http://arxiv.org/abs/1907.11692

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter* (arXiv:1910.01108). arXiv. http://arxiv.org/abs/1910.01108

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. http://arxiv.org/abs/1706.03762

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation* (arXiv:1609.08144). arXiv. http://arxiv.org/abs/1609.08144

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., & Toutanova, K. (2019). *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions* (arXiv:1905.10044). arXiv. http://arxiv.org/abs/1905.10044

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018). *Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering* (arXiv:1809.02789). arXiv. http://arxiv.org/abs/1809.02789

Rajpurkar, P., Jia, R., & Liang, P. (2018). *Know What You Don't Know: Unanswerable Questions for SQuAD* (arXiv:1806.03822). arXiv. http://arxiv.org/abs/1806.03822

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text* (arXiv:1606.05250). arXiv. http://arxiv.org/abs/1606.05250

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations* (arXiv:1802.05365). arXiv. http://arxiv.org/abs/1802.05365

Sennrich, R., Haddow, B., & Birch, A. (2016). *Neural Machine Translation of Rare Words with Subword Units* (arXiv:1508.07909). arXiv. http://arxiv.org/abs/1508.07909

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding* (arXiv:1804.07461). arXiv. https://doi.org/10.48550/arXiv.1804.07461