**Summary of the Finalized Papers**

**Paper 1:**  
Benjamin M. Ampel et al. [1] (2024) proposed the TAISR framework intending to exploit LLMs efficiently for text-type tasks and to quell the high operational costs on one side and on the other side operative issues with evaluation. Models such as FinBERT (F1: 91.34%), Pegasus (ROUGE-1: 0.443), and GPT-NeoX served best on the money news, social media posts, and Wendy's Tweets datasets. The framework under study is less demanding concerning monetary aspects but desperately needs new-developed methods to evaluate it properly while also being a lot less demanding computationally.  
**[1.1]** Provides a comprehensive overview of the architecture, training, and applications of large language models (LLMs).  
**[1.2]** An updated and extended survey covering recent advances, challenges, and ethical concerns in LLM development.  
**[1.3]**Analyzes the societal, industrial, and academic impact of LLMs, along with potential use cases.  
**[1.4]**Discusses how LLMs can enhance business intelligence workflows through automation and smarter decision-making.

**Paper 2:  
Dominic Widdows et al. [2] (2024)** proposed a **Prompt Pattern Catalog** with 26 reusable templates in 6 categories to improve prompt engineering for LLM tasks like summarization and classification. It enhances **clarity, consistency, and transferability** without relying on fixed datasets, and showed better performance than random prompts. While effective, it **lacks empirical benchmarks** and **automated pattern selection tools**.  
**[2.1]**Introduces a structured catalog of 26 reusable prompt patterns to improve prompt design and task performance in ChatGPT.  
**[2.2]**Surveys various prompt engineering strategies across NLP tasks, analyzing their effectiveness and categorizing recent methods.  
**[2.3]** Proposes foundational ideas for effective prompt crafting in LLMs, focusing on techniques, challenges, and future directions.  
**[2.4]** Provides a practical, systematized review of prompt engineering techniques and their real-world applications using LLMs.

**Paper 3:  
Jason Wei et al. [3] (2022)** introduced **Chain-of-Thought (CoT) prompting** to improve reasoning in LLMs through step-by-step problem solving. Using **GPT-3** on datasets like **GSM8K, CommonsenseQA, and AQuA**, accuracy improved significantly (e.g., **17.7% to 58.1% on GSM8K**). It enables complex reasoning without extra training but struggles with **smaller models** and relies on **handcrafted prompts**.  
**[3.1]** Step-by-step prompting is introduced as a way to really allow complex reasoning for advanced tasks in LLMs.  
**[3.2]** Approaches using Socratic questioning to lead the LLMs into a deeper and more transparent reasoning path.   
**[3.3]** Studies impact of zero-shot CoT prompting on the improvement of diagnostic reasoning in medical visual question answering.  
**[3.4]** Proposes to automatically improve zero-shot prompts so that LLMs might better reason without fine-tuning.

**Paper 4:  
Yizhong Wang et al. [4] (2022)** studied instruction fine-tuning using the **UnifiedSKG framework** to evaluate performance across **21 NLP tasks**. Using models like **T5-base and T5-large** on datasets such as **WebQuestions, WikiSQL, and SParC**, UnifiedSKG outperformed task-specific models in **generalization and transfer**. While it showed **robustness and scalability**, limitations included **sensitivity to prompts** and **high resource usage**.  
**[4.1]**Reviews recent LLMs used for code generation, highlighting models, benchmarks, and future research directions.  
**[4.2]**Analyzes how LLMs reflect or amplify biases in sentiment analysis across different financial companies.  
**[4.3]** Enhances the Fama-French model by integrating fine-tuned LLM-based sentiment analysis for better market predictions.  
**[4.4]**Demonstrates how LLMs can accurately capture and analyze sentiment in financial texts for decision-making.  
**[4.5]** Provides a structured review of interpretable LLM techniques for assessing credit risk in finance.

**Paper 5:  
Hanqing Zhang et al. [5] (2024)** proposed the **Residual Memory Transformer (RMT)** to enable fine-grained, controllable text generation in LLMs like GPT-2 without altering the base model. Tested on **CommonGen, C2Gen, and SST-5**, RMT outperformed baselines like **DisCup and DEXPERT**, offering **efficiency and minimal parameter tuning** (~16%). Limitations include incompatibility with **closed-source models** and **occasional commonsense errors**.  
**[5.1]** Surveys techniques and challenges in steering LLM outputs for attributes like tone, style, and content.  
**[5.2**Examines controllable text generation through the lens of causal inference, highlighting control mechanisms.  
**[5.3]** Introduces a reinforcement-based method to unlearn or enforce specific attributes in generated text.  
**[5.4]** Proposes CTRLEval, a metric to evaluate controlled text generation without needing reference outputs.

**Paper 6:**  
Zhiqiang Song et al.[6] (2023) have proposed BERT-LongDocuments, a model to improve the processing of long documents in various NLP applications such as information retrieval and topic classification. Pre-training of the BERT-LongDocuments were carried out on PubMed and ArXiv, while evaluation was carried out on IMDb, 20NewsGroup, ArXiv, and PubMed, outperforming all competing models. Although it models long inputs well, somewhat of a limitation also exists with regard to very long texts, and it also requires very heavy computational resources for pre-training.  
**[6.1]** Proposes an online plagiarism detection system with deep learning methods such as Doc2Vec, Siamese LSTM and CNN of high accuracy.

**[6.2]** An end-to-end model for entity-level relation extraction through multi-instance learning in order to increase performance.   
**[6.3]** Describes the deep learning approaches of JNLP teams on legal text processing for the COLIEE 2021 competition.  
**[6.4]** Transfer learning is applied for image analytics in legal document review processes, thus supporting multimedia eDiscovery.

**Paper 7:**  
Cingillioglu et al.[7] (2023) used the AI chatbot Sydn-e in an RCT with 1193 global participants to study university choice factors, emphasizing positive eWOM. Five hypotheses were supported, while ease of admission, prior knowledge, and collaboration had little effect. The approach improved efficiency, reduced bias and cost, but faced limitations like AI bias and restricted use in low-tech settings.   
**[7.1]**Tutor CoPilot: Human-AI system that raises the tutoring efficacy and scalability.  
**[7.2]**AI in Education Trends: The AI technologies forging the path of personalized learning.  
**[7.3]**AIIA Framework: AI assistant facilitating adaptive and interactive learning.  
**[7.4]**Artificial Intelligence for Personalized Learning: It increases engagement, yet ethical constraints come into play.

**Paper 8:**  
Ayman Asad Khan et al. [8] (2024) proposed an end-to-end RAG system pipeline using PDFs and GPT/Llama models to improve accuracy and reduce hallucinations without using benchmark datasets. While enhancing contextual retrieval, challenges include complex infrastructure, vector databases, and PDF extraction errors; the study shares practical insights and open-source Python code.  
**[8.1]**Khan et al. (2024): Proposed a workable RAG pipeline with GPT/LLaMA using PDFs to enrich context and share open-source code.  
**[8.2]**Raina & Gales (2024): Improved RAG retrieval by segmenting documents into atomic units and synthetic questions.  
**[8.3]**Wang et al. (2024): Hosted the first workshop on R³AG and aimed to improve the reliability of RAG systems.  
**[8.4]**Su et al. (2024): Proposed DRAGIN, a dynamic RAG technique that fine-tunes retrieval in response to evolving LLM needs.

**Paper 9:**  
Yuping Wu et al. [9] (2024) proposed EXTABS, a unified summarization framework combining extractive and abstractive methods with BART and PEGASUS to improve coherence and efficiency. Tested on CNN/DailyMail, Reddit, and PubMed, it outperformed baselines by reducing duplication, errors, and training cost through a novel saliency mask and joint encoder-decoder training. However, it may still face challenges in handling extremely long documents and requires further optimization for diverse domains.  
**[9.1]**Abstractive summarization could install extractive summarization capabilities to its front end via innovative inference methods.  
**[9.2]**SumHiS improves extractive summarization by making use of latent document structures and clustering techniques.   
**[9.3]**When extractive summarization is treated as text matching, one improves the semantic relevance of summaries.  
**[9.4]**Using discourse-aware attention and rhetoric structures theory enhances the coherence of extractive summaries.

**Paper 10:**  
Norman Meuschke and Bela Gipp [10] (2013) reviewed plagiarism detection methods like fingerprinting and stylometry without proposing a specific model or dataset. Their study highlights strengths in detecting verbatim plagiarism and emerging techniques for paraphrased cases, while noting the need for human oversight and lack of experimental validation.  
**[10.1]** Semantic concept pattern analysis detects paraphrased and translated plagiarism.  
**[10.2]**An integrated plagiarism detection system checks text, citations, images, and math.  
**[10.3]**Online courses open up academic integrity concerns, and there are strategies to deter cheating.  
**[10.4]**Language independent implies that the plagiarism detection software analyses citations and images.

**Paper 11:  
R. Krishna et al. [11] (2025)** examine how large language models generate reworded academic content that evades standard plagiarism tools. Their expert-backed review reveals widespread disguised copying and urges smarter detection methods alongside stronger institutional policies. They emphasize the need for systems that detect not just surface-level similarity but also deeper semantic imitation.  
**[11.1]** Highlights how traditional plagiarism tools are not able to detect AI-generated content.  
**[11.2]** Proposes stylometric and neural model-based methods to detect AI-written texts.  
**[11.3]** Discusses the ethical concerns of AI misuse for academic writing and needs for policy reform.  
**[11.4]** Reviews current detection methods pointing out their weaknesses vis-à-vis generative AI.

**Paper 12:  
T. Saglam et al. [12] (2025)** examine how AI-based code changes make it hard for current plagiarism tools to catch copied software. They find that many tools fail when code is auto-modified by AI. This raises concerns about fairness in coding assignments and shows the need for better detection systems. Their study calls for stronger academic policies to handle AI use in programming.  
**[12.1]** Examines GitHub Copilot’s code and its risks for secure, original contributions.  
**[12.2]** Surveys how code representation learning aids in understanding and detecting code similarity.  
**[12.3]** Identifies key challenges in detecting AI-assisted software plagiarism in educational settings.  
**[12.4]** Reviews core plagiarism detection techniques and their relevance in the AI era.

**Paper 13:  
M.-T. C. Evans et al. [13] (2024)** examine the potential of semantic text analysis to enhance transformer-based models for relation extraction. Their research shows improved understanding of complex textual relationships, strengthening the foundation for plagiarism detection and semantic similarity tasks. The findings suggest deeper semantic insights can significantly boost accuracy in identifying reworded or conceptually similar content.  
**[13.1]** Uses pretrained transformers to propose a semantic-aware method for plagiarism detection.  
**[13.2]** Integrates semantic graphs for deeper context, improving text similarity scoring.  
**[13.3]** Evaluates the accuracy of several text embedding methods for plagiarism detection.  
**[13.4]** Effectively detects semantic similarity in academic writing using neural networks.

**Paper 14:  
K.-U. Sattler et al. [14] (2024)** focus on efficient similarity search techniques using vector sets, laying foundational work for scalable plagiarism detection. Their approach supports fast and accurate retrieval, which is highly relevant for modern vector-based text similarity in academic integrity systems. The study remains influential for current applications using nearest-neighbor and vector embedding techniques.  
**[14.1]** Introduces scalable similarity search strategies for preserving academic integrity.  
**[14.2]** Uses FAISS-based vector matching for large-scale text similarity detection.  
**[14.3]** Explores vector databases to detect plagiarism in academic submissions.  
**[14.4]** Applies approximate nearest neighbor search to improve educational applications.

**Paper 15:  
J. Lee et al.[15] (2024)** introduce PlagBench, a benchmark for evaluating how large language models both generate and detect plagiarism. Their work shows how LLMs can unintentionally create plagiarized content and how detection models struggle to identify it, emphasizing the dual role of AI in this context. They call for standardized evaluation to ensure future detection tools remain effective.  
**[15.1]** Evaluates transformer models in plagiarism detection through experimental analysis.  
**[15.2]** Examines the ethical implications of LLM use in academic environments.  
**[15.3]** Benchmarks popular plagiarism detection tools against LLM-generated text.  
**[15.4]** Compares LLM behaviors in both plagiarism generation and detection scenarios.

**Paper 16:** *Meuschke* and co-authors benchmark ten open-source tools that pull structured data from academic PDFs. They evaluate tools on tasks like extracting metadata, references, tables, and full content using the DocBank dataset. **GROBID** stands out in metadata and content tasks, while **Adobe Extract** leads in table recognition. Still, all tools struggle with complex parts like equations, lists, and footers. The authors suggest creating a shared evaluation framework. They also recommend hybrid models and synthetic data to improve accuracy. This work makes it easier for researchers to choose tools and supports future improvements in academic PDF parsing.

**[16.1]** Lewis et al. (2020) propose RAG, a model combining retrieval with generation to reduce hallucination and improve factual accuracy in knowledge-intensive NLP tasks.  
**[16.2]** Lopez (2008–2024) presents GROBID, an open-source tool for extracting structured bibliographic data from scholarly PDFs.  
**[16.3]** Kamradt (2024) introduces semantic chunking techniques to enhance document retrieval by splitting text based on meaning and hierarchy.  
**[16.4]** Dušek et al. (2023) enhance domain-specific retrieval by fine-tuning retrievers using Natural Language Inference (NLI) techniques.  
**[16.5]** Es et al. (2023) introduce RAGAS, a framework for automated evaluation of retrieval-augmented generation outputs across multiple quality dimensions.

**Paper 17:**In this work, Patrick Lewis and team introduce RAG—a smarter way for language models to pull real info while generating text. RAG combines BART’s writing skills with a document retriever (DPR) to fetch facts from Wikipedia as it writes. Two versions—RAG-Token and RAG-Sequence—control when retrieval happens. Compared to older models, RAG gives clearer, more accurate answers, especially in question answering and fact-checking. What’s cool is the retriever learns to find useful—not just exact—matches. Human evaluations backed its strong performance. This approach sets a new direction for building AI systems that are more reliable and grounded.

**[17.1]**  
Meuschke et al. (2023) benchmark various PDF extraction tools using a solid multi-task and multi-domain evaluation framework tailored for academic documents.

**[17.2]**  
Lopez (2009) introduces GROBID, a tool that smartly extracts bibliographic and term data from scholarly PDFs using machine learning.  
**[17.3]**  
Li et al. (2020) present DocBank, a large dataset designed to help train models for understanding the layout of academic documents.  
**[17.4]**  
Tkaczyk et al. (2015) build CERMINE, a system that automatically pulls structured metadata from research papers using advanced parsing techniques.  
**[17.5]**  
Lafferty, McCallum, and Pereira (2001) propose Conditional Random Fields (CRFs), a statistical model great for labeling and segmenting sequence data like text.  
**[17.6]**  
Councill, Giles, and Kan (2008) develop ParsCit, an open-source tool that uses CRFs to accurately parse and label citation strings in academic writing

**Paper 18:***“Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,”* Patrick Lewis and his team propose a smarter way to make language models more reliable and accurate. They introduce **RAG (Retrieval-Augmented Generation)**—a hybrid system that combines the generative power of BART with the retrieval abilities of DPR (Dense Passage Retriever). Instead of relying on whatever static knowledge a language model was trained on, RAG actively pulls information from a dense index of Wikipedia during generation. This makes it **less likely to hallucinate** facts and much better at grounding its answers in real, verifiable content. The authors present two versions—**RAG-Sequence** and **RAG-Token**—which differ in *how* they pull in information mid-generation. When tested on tasks like question answering, fact-checking, and generating new questions, RAG outperformed standard models. Human evaluations also confirmed that its responses were **more specific and accurate**. What’s especially cool is that the retriever learns to fetch not just exact matches but genuinely useful info. This paper really pushes the boundaries of what **trustworthy, updatable NLP systems** can look like and is a solid step toward making AI more dependable.  
**18.1]**  
Lewis et al. (2020) propose RAG, a model that combines search (retrieval) and text generation to improve performance on complex NLP tasks.  
**[18.2]**  
Karpukhin et al. (2020) introduce Dense Passage Retrieval (DPR), a method for finding relevant passages using deep learning for open-domain Q&A.  
**[18.3]**  
Guu et al. (2020) present REALM, a language model that learns by retrieving real-world documents while training, boosting factual accuracy.  
**[18.4]**  
Raffel et al. (2019) create T5, a powerful text-to-text model that treats all NLP problems as text generation tasks and nails transfer learning.  
  
**[18.5]**  
Lewis et al. (2019) design BART, a flexible model that learns to fix corrupted text and works great for translation, summarization, and more.

**Paper 19:***S2ORC*, introduced by *Lo et al.*, is a giant, structured dataset of over 81 million academic papers for NLP research. It includes full text for 8.1 million open-access papers, along with helpful extras like citation links and figure/table references. Built using tools like GROBID and ScienceParse, it also clusters different versions of the same paper. They trained a custom model, S2ORC-SciBERT, on this dataset—and it performed great on scientific NLP tasks. Manual checks showed strong data quality and accuracy. S2ORC powers applications like summarization and citation analysis. It's now a key resource behind tools like CORD-19.

**[19.1]**  
K. Lo et al. (2020) introduce **S2ORC**, a large, diverse, and open corpus of scientific papers for NLP research, supporting tasks like citation analysis and information extraction.

**[19.2]**  
L. Lu et al. (2020) present **CORD-19**, a COVID-19-focused open research dataset to support text mining and retrieval applications during the pandemic.

**[19.3]**  
L. U. Wang et al. (2020) adapt **CORD-19** from S2ORC and release it on Kaggle to demonstrate practical use of structured scientific data.

**[19.4]**  
C. Beltagy et al. (2019) propose **SciBERT**, a transformer-based model trained on scientific text, outperforming BERT on scientific NLP tasks.

**Paper 20:**  
This paper tackles AI-generated plagiarism, especially from models like ChatGPT and Gemini, which make it easy to cheat. The authors review detection methods like classifiers, zero-shot detectors, and watermarking—but note that paraphrasing or smart prompting often beats them. Current tools aren’t reliable and lack consistent benchmarks. Instead of chasing perfect detectors, the authors argue for changing how we teach and assess. They suggest designing assignments that AI can't easily solve and teaching students to use AI responsibly. The message is clear: better education and ethics—not stricter bans—are the long-term solution to AI misuse in academia.

**[20.1] V. Sadasivan et al. (2023):** Investigates the reliability of detecting AI-generated text using various detection models.  
**20.2] E. A. Oliveira et al. (2025):** Analyzes student writing to detect AI involvement using stylometric features, raising concerns about academic misconduct.  
**[20.3] S. T. Sağlam and L. Schmid (2025):** Evaluates plagiarism detection tools against AI-generated and obfuscated software code to assess integrity risks.  
**[20.4] S. Leaton Gray et al. (2025):** Discusses how AI-enabled digital cheating reshapes higher education and proposes ethical pedagogical responses.  
**[20.5] J. Lee et al. (2024):** Introduces PlagBench to assess how LLMs can both generate and detect plagiarism, highlighting their dual nature.