

Article

Trend Analysis of Large Language Models through a Developer Community: A Focus on Stack Overflow

Jungha Son  and Boyoung Kim * 

Seoul Business School, aSSIST University, Seoul 03767, Republic of Korea; jhson@stud.assist.ac.kr

* Correspondence: bykim2@assist.ac.kr; Tel.: +82-10-4046-2428

Abstract: In the rapidly advancing field of large language model (LLM) research, platforms like Stack Overflow offer invaluable insights into the developer community's perceptions, challenges, and interactions. This research aims to analyze LLM research and development trends within the professional community. Through the rigorous analysis of Stack Overflow, employing a comprehensive dataset spanning several years, the study identifies the prevailing technologies and frameworks underlining the dominance of models and platforms such as Transformer and Hugging Face. Furthermore, a thematic exploration using Latent Dirichlet Allocation unravels a spectrum of LLM discussion topics. As a result of the analysis, twenty keywords were derived, and a total of five key dimensions, "OpenAI Ecosystem and Challenges", "LLM Training with Frameworks", "APIs, File Handling and App Development", "Programming Constructs and LLM Integration", and "Data Processing and LLM Functionalities", were identified through intertopic distance mapping. This research underscores the notable prevalence of specific Tags and technologies within the LLM discourse, particularly highlighting the influential roles of Transformer models and frameworks like Hugging Face. This dominance not only reflects the preferences and inclinations of the developer community but also illuminates the primary tools and technologies they leverage in the continually evolving field of LLMs.

Keywords: large language model; Transformer; Hugging Face; Stack Overflow; developer community



Citation: Son, J.; Kim, B. Trend Analysis of Large Language Models through a Developer Community: A Focus on Stack Overflow. *Information* **2023**, *14*, 602. <https://doi.org/10.3390/info14110602>

Academic Editors: Polona Tominc and Maja Rožman

Received: 18 September 2023

Revised: 3 November 2023

Accepted: 3 November 2023

Published: 6 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, the introduction of OpenAI's ChatGPT has significantly heightened interest in large language models (LLMs) [1]. The evolution and transformation of LLMs have reshaped the artificial intelligence landscape [2,3], facilitating substantial advancements, especially in the realm of natural language processing, where large-scale datasets have been utilized to explore a myriad of research inquiries, contributing to the progression of the field [4]. Concurrently, neural networks have also considerably advanced.

Previous research has introduced efficient deep neural network models for various applications, underscoring the potential of neural networks in handling complex tasks and large datasets [5]. These advancements have been additionally bolstered with the advent of other pioneering models, such as Google's PaLM and Meta's LLaMA [6,7]. In this context, the developers note that the associated models alter various aspects of artificial intelligence, ranging from natural language processing to complex tasks such as code generation and completion [8–10].

Language models have undergone notable developmental phases since the inception of computational linguistics. The earlier models were primarily rooted in rule-based systems, featuring handcrafted rules and lexicons to model linguistic phenomena [11]. While these models were pivotal in initial explorations of machine language processing, they often could not achieve the scalability and flexibility necessary to capture the nuances of natural language [12,13]. The emergence of neural networks in the early 21st century significantly impacted language modeling. Among these, the neural probabilistic language models stood out, utilizing the power of distributed representations to model sequential

data [14]. Despite their promise, these models faced difficulties in handling long-term dependencies due to the inherent limitations of recurrent networks [15].

The field underwent a monumental shift with the introduction of transformer architecture, pioneered by Vaswani et al. [16] in their seminal paper, “Attention Is All You Need”. Eschewing recurrence altogether, this architecture relies on a self-attention mechanism, enabling the transformers to weigh input tokens in a sequence and effectively capture the context, regardless of the positional distance.

Building on this architecture, several influential models emerged. BERT (Devlin et al., 2018), for instance, leveraged bidirectional transformers to understand the context from both the left and right of a token, pioneering a pre-training and fine-tuning approach. This methodology, where models are first trained on a vast corpus and then fine-tuned for specific tasks, became a staple in subsequent LLM developments [17]. After BERT, the GPT series, particularly GPT-2 and GPT-3, pushed the boundaries further. These models, characterized by their autoregressive nature, demonstrated unparalleled capabilities in a range of tasks, from language generation to translation and beyond [18].

The ongoing advancements and integration of theoretical insights with computational capability highlight the potential of LLMs to transform artificial intelligence. Alongside this technological evolution, notable dynamism within the developer community has become apparent [19]. As these tools and models emerged, developers worldwide collaborated to adapt, comprehend, and refine them.

As the field of LLM has developed, the community surrounding it has concurrently progressed. Platforms such as GitHub, arXiv, and specialized forums have experienced a surge in activity, becoming hubs for LLM discussion and collaboration [20]. A pivotal transformation has been the democratization of LLM access and development. LLMs are no longer dominated by select institutions and corporations. Independent developers and researchers are now providing significant contributions, aided by major organizations that release pre-trained LLM models. This allows a wider audience to adapt them for specific applications without necessitating substantial computational resources [21].

The collaborative spirit within the developer community has catalyzed an increase in open-source projects, fostering shared learning and collective development. In particular, Stack Overflow is worthy of attention, having established itself as an indispensable resource for the LLM community. Recognized as the world’s largest developer community, its significance transcends mere troubleshooting, acting as an indicator of evolution in the technology sector. In the context of LLMs, Stack Overflow offers a wealth of insights into real-world applications and the accompanying challenges. Research into Stack Overflow, exemplified by the work of Barua et al. [22], illuminates how developers interact with the platform, uncovers the challenges they encounter, and provides a valuable resource for the development of nascent technologies like LLMs.

As Ford et al. [23] articulate, platforms such as these effectively unite research and pragmatic applications of software engineering, affording developers access to cutting-edge solutions. This interconnectedness is notably evident in the perpetually evolving domain of LLMs, wherein conventional documentation might not contemporaneously align with challenges and solutions during practical implementation [17,18,24].

Despite the attention LLMs have attracted and the evident importance of platforms like Stack Overflow in illustrating the technology landscape, there is a notable lack of in-depth discourse analysis. Investigating this aspect is vital, as understanding the real-world application and discourse around LLMs can connect theoretical advancements with practical implementations [25,26]. Previous studies have delved into diverse topics, such as predicting the acceptability of answers in specific programming languages [27], understanding refactoring discussions [19], and exploring the use of technologies like Apache Spark [28]. Additionally, research to comprehend discussions around deep learning frameworks has been undertaken, as evidenced by the mining and comparing of discussions on platforms like Stack Overflow and GitHub [20]. However, the discourse and trends specific to LLMs on Stack Overflow remain uncharted. This research endeavors to bridge

this gap by providing a meticulous analysis of LLM-related discussions and trends on Stack Overflow and, thereby, offering a novel contribution that is distinct from previous studies.

This study aims to explore the technology and development trends of LLMs based on experts' discussions and opinions using an open data platform. The research questions addressed are as follows: "How do LLM-related Tags on Stack Overflow exhibit usage frequency and evolving trends, and what significance do Tags representing new technologies and frameworks hold?" and "How do dominant keywords and themes shape the LLM-related discussions on Stack Overflow, and what insights can be drawn from the interplay and interconnectedness of these terms about the evolving narrative within the developer community?" Interactions on Stack Overflow offer invaluable insights into the real-world challenges, adaptations, and innovations associated with LLMs. The Tags, questions, and thematic discussions on the platform provide insights into the prevailing tools, frameworks, and practices within the developer community [29]. As LLMs are integrated into diverse applications, the big data trend analysis of the discourse has become pivotal, offering invaluable suggestions to researchers, developers, and tool creators.

Consequently, this study delves into LLM discussions on Stack Overflow, highlighting the practical implementation challenges encountered by developers. It explores the need to foster a dialogue between technological advancements and practical usability in the LLM domain while simultaneously paving a pathway for future research on the evolving challenges faced by artificial intelligence developers.

2. Methods

2.1. Data Collection and Pre-Processing

As depicted in Figure 1, the methodology encompassing our study provides a systematic and in-depth approach to exploring the interactions and queries of developers on the Stack Overflow platform. Specifically, it concentrates on the discussions and advancements relevant to LLMs post 2017.

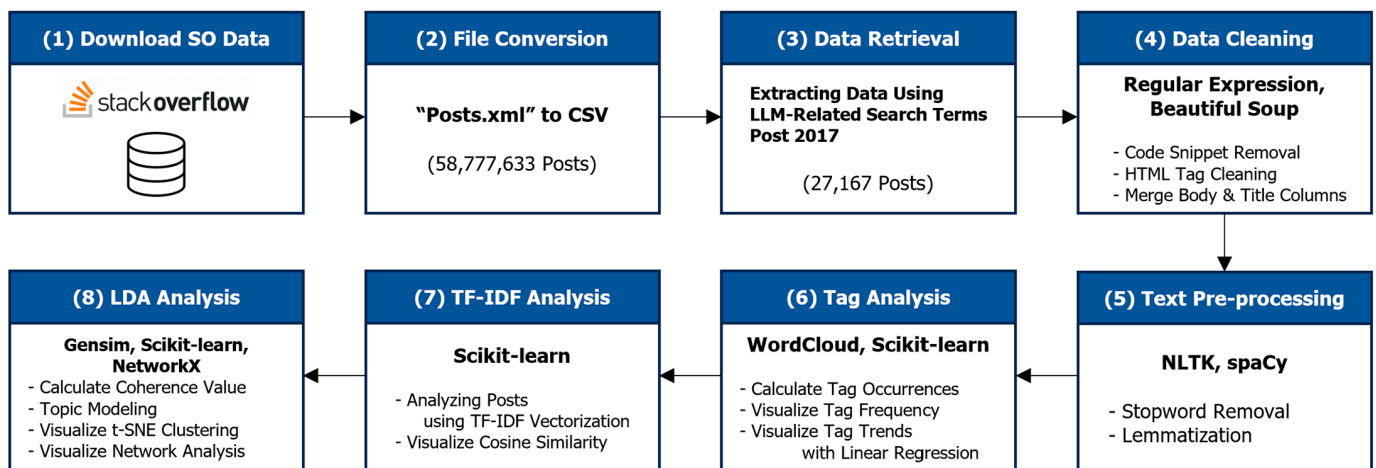


Figure 1. Overview of methodology of our study.

The data dump from Stack Overflow, made available under the Creative Commons BY-SA 4.0 license, has been released by Stack Exchange in an XML format, thereby facilitating a wide spectrum of analyses for researchers. This study utilized the dataset archived on 30 July 2023, which comprises multiple XML files, including posts, comments, Tags, badges, post history, post links, users, and votes. With a focus on analyzing the developers' questions and answers, the Posts.xml file was leveraged. For the purpose of our analysis, the Posts.xml file was parsed into a CSV (Comma-Separated Values) format [30,31].

The insight-rich dataset sourced from Stack Overflow necessitated a comprehensive pre-processing strategy to align with the analytical goals of the study [32]. Priority was given to specific columns: Id, PostTypeId, CreationDate, Body, Title, and Tags. While

the code embedded within Stack Overflow posts using `<code>` Tags can be contextually relevant, it was considered extraneous for textual analysis and, thus, was purged using regular expressions. Additionally, any remaining HTML artifacts in the Body column were sanitized using the BeautifulSoup library. To maintain uniformity, commas present in the Body, Title, and Tags columns were substituted with spaces. Considering the dataset's magnitude, this pre-processing procedure was executed in portions, with each refined chunk saved in a distinct directory.

To make the analysis relevant, we confined the timeframe to data from 2017 onwards. This choice was influenced by pivotal developments in 2017, specifically the introduction of the Transformer architecture and the self-attention mechanism from Google. These breakthroughs signified a transformative period in NLP, igniting widespread research on LLMs [16].

After initial data cleansing, the focus transitioned to further refining the dataset, with a specific emphasis on discussions pertaining to LLMs and related technological advancements [33]. A curated list of keywords, emblematic of current strides in NLP, was developed. This list included terms such as “transformer”, “large language model”, and “gpt” and acknowledged recent innovations from organizations like “OpenAI”. Moreover, concepts like “self-attention” and “fine-tuning” and emerging trends such as “prompt engineering” and “few-shot learning” were integrated into this keyword list. By utilizing regular expressions, the entries in the “Body” and “Title” columns that featured these keywords were retained, shaping the dataset for more focused analysis.

To delve into the textual data, several refinement stages were undertaken to enhance its readiness for analysis. Common English words, or stop words, were identified and excluded using the NLTK library, prioritizing terms with a higher analytical value. Subsequently, normalization was executed, which involved converting the entire text corpus to lower case and excising non-alphabetical symbols, resulting in a refined, word-focused dataset [34]. The Gensim library was utilized to identify bigrams, enabling the detection of nuanced linguistic constructs. In the final refinement stage, SpaCy was used for lemmatization, ensuring that all the word variants were mapped to their foundational form [35,36]. These meticulous steps are crucial in enhancing the dataset's integrity and laying a robust foundation for subsequent analytical endeavors.

2.2. Research Design and Process

This section elucidates the methodology and techniques employed. We addressed the first research question, “How do LLM-related Tags on Stack Overflow exhibit usage frequency and evolving trends, and what significance do Tags representing new technologies and frameworks hold?”. In deciphering the discussions surrounding the LLM on Stack Overflow, ensuring analysis integrity was paramount. A key component involved the reliable selection and interpretation of relevant Tags from the “Tags” column using Python. To maintain consistency in the data, entries without Tags were filtered out. This method aimed to embed individual Tags within the wider discourse, highlighting their significance over time. Exponential modeling, particularly with log-transformed data, facilitated the clear representation of Tag trends. Additionally, utilizing word cloud visualizations enabled a succinct and intuitive display of the prominence and frequency of the discussed Tags.

To address the subsequent research question, “How do dominant keywords and themes shape the LLM-related discussions on Stack Overflow, and what insights can be drawn from the interplay and interconnectedness of these terms about the evolving narrative within the developer community?”, textual analysis was performed. This approach was utilized to extract and understand thematic layers and semantic connectivity among terminologies within the discussions.

Shifting the focus to the textual content of Stack Overflow posts, the “Title” and “Body” were combined into a single entity, aiming for a comprehensive understanding of the users' discussions about LLMs. The TF-IDF (Term Frequency–Inverse Document Frequency) technique was employed using this combined data, involving more than

the term frequency [37]. It emphasized terms that were uniquely significant to specific discussions. By aggregating the TF-IDF scores across all the discussions, the terms with general importance were brought to the forefront. Additionally, the cosine similarity measure played a pivotal role in uncovering semantic links between these critical terms, illuminating potential associations and correlations in the discussions [38,39].

In response to the complexities arising from the vast consolidated textual content, conventional reading methods were insufficient. Thus, Latent Dirichlet Allocation (LDA) was utilized as a means to discern underlying themes and narratives within the data [40]. Based on initial data insights, LDA was configured to extract five dominant topics. To enhance the visualization and interpretation of these topics, t-SNE (t-Distributed Stochastic Neighbor Embedding), known for transforming high-dimensional data into a two-dimensional space, was employed [41]. t-SNE is valued for its capability to create visually interpretable models, which render the abstract outputs of LDA into more tangible, comprehensible visuals.

Lastly, network analysis was incorporated to complement the thematic insights derived from LDA and present a clearer image of keyword interrelations. The networkx library was instrumental in this phase, presenting keywords as nodes and their thematic interconnections as edges. The visualization's spring layout was opted for, ensuring the articulate presentation of keyword groupings and overlaps [42,43].

3. Results

3.1. Overview of the Dataset

The analysis utilized the comprehensive Stack Overflow data dump, offering an inclusive overview of developer interactions on the platform. This vast dataset encompasses a wide range of topics, discussions, and user engagements, as evidenced by the multitude of posts, questions, Tags, and active users. Data on LLM discussions since 2017 were selectively analyzed. This subset highlights the community's engagement with LLMs, as shown by the number of LLM-related questions, Tags, and interactions by distinct users. Table 1 presents a comparative view of the entire dataset against the LLM-related subset, enabling an understanding of the dataset's scope and diversity.

Table 1. Statistics about the collected data.

Item	Entire Dataset	LLM-Related Subset (Since 2017)
Number of posts	58,777,633	27,167
Number of questions	23,719,810	16,769
Number of answered questions	20,291,340	11,180
Number of accepted answers	12,057,994	5257
Number of distinct Tags	64,785	5400
Number of distinct users	6,112,508	20,482
Average number of Tags per question	2.97	3.40
Average number of answers per question	1.47	0.87

3.2. Linear and Word Cloud Analysis of the Tags

Within the expansive domain of Stack Overflow, Tags are instrumental, guiding users to germane discussions and demarcating the primary thematic focal points of the content. The analytical examination of the dataset unveiled the Tags that are frequently invoked, particularly those that resonate with LLMs.

As shown in Figure 2, the actual monthly frequencies of questions linked to these Tags are represented by blue solid lines. The red dashed lines, on the other hand, signify the linear fit of the log-transformed data. This log transformation is crucial for capturing the exponential nature of discussions around these topics, providing a clearer perspective on their growth trajectories. The slope of the linear fit, represented by the coefficient values from a linear regression, encapsulates the monthly trend of discussions for each Tag.

A higher coefficient indicates a steeper incline, signaling a more rapid monthly increase in discussions about that Tag.

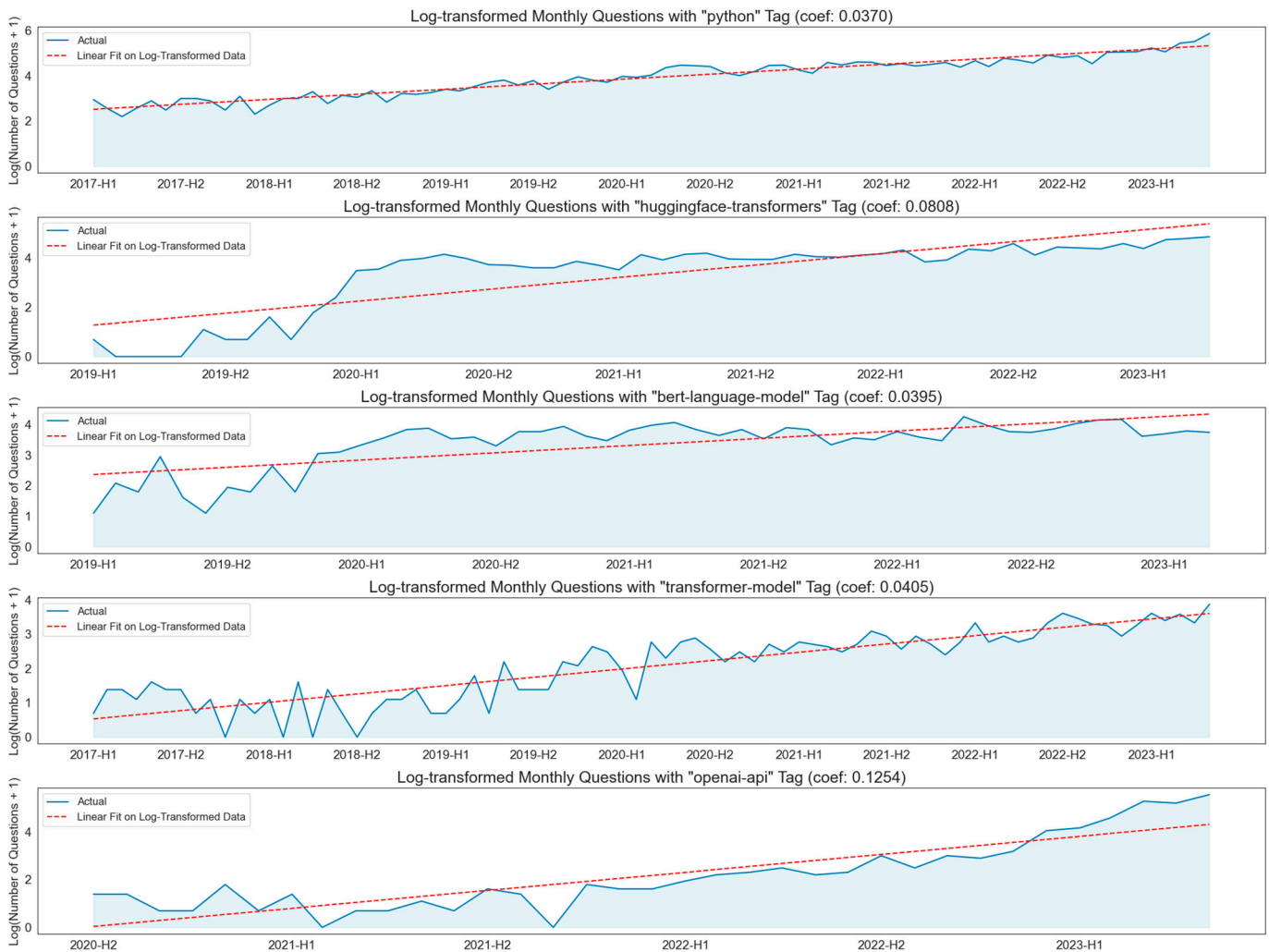


Figure 2. Log-transformed trends of key Tags.

Delving into the specifics, the “huggingface-transformers” Tag shows a particularly sharp incline, hinting at its rising influence and rapid adoption among developers. This reflects the burgeoning popularity of the Hugging Face’s Transformers library in the LLM domain. The “openai-api” Tag, with the steepest coefficient among the observed Tags, signifies the surging interest in offerings from OpenAI, notably when it introduced pioneering models and tools. The “python” Tag demonstrates consistent and steady growth, emphasizing its foundational role in LLM-related developments. Similarly, the Tags “bert-language-model” and “transformer-model” exhibit growth patterns that underscore the continued relevance and foundational significance of BERT and transformer architectures in NLP, respectively. The insights garnered from Figure 2 emphasize the heterogeneous engagement levels and growth associated with various LLM Tags on Stack Overflow. The comparative analysis of these growth trajectories offers insights into the shifting focal points of discussion, illuminating both the foundational pillars and emergent areas of interest within LLM discourse.

Figure 3 presents a word cloud that underscores the salience of the selected Tags. Among them, “python”, “huggingface-transformers”, “huggingface”, “bert-language-model”, “openai-api”, and “chatgpt-api” are distinct. The “python” Tag, with its pronounced visibility, corroborates its central position in LLM development and deployment.

Table 2. Top 20 dominant terms using TF-IDF vectorization.

Rank	Term	TD-IDF Score
1	use	1192.12
2	model	1165.78
3	transformer	887.52
4	error	711.59
5	code	682.28
6	try	662.08
7	bert	560.09
8	work	553.53
9	python	552.85
10	data	541.80
11	file	537.94
12	like	530.44
13	train	528.98
14	huggingface	513.91
15	follow	468.42
16	need	452.72
17	run	446.52
18	want	442.62
19	openai	427.15
20	api	424.09

Heatmap analysis was conducted to confirm the semantic interaction between the most important terms extracted through TF-IDF vectorization. The heatmap analysis results demonstrate that a pivotal attribute is the cosine similarity value, wherein terms with augmented semantic consistency or those frequently bundled in user queries are emphasized.

As illustrated in Figure 4, the term “huggingface” is similar to “transformer” and “bert”. This alludes to recurrent dialogues centered around Hugging Face’s transformer libraries and the BERT model, highlighting their criticality in the LLM domain. Furthermore, a moderate level of affinity between “data” and “file” intimates their recurrent association, presumably within the ambit of data processing or file operations. An intriguing parallel between “openai” and “api” is also depicted, potentially suggesting discussions primarily revolving around the functionalities of OpenAI’s API. These analytical undertakings yield invaluable insights, enabling both researchers and practitioners to delineate the focal points of LLM discourse on Stack Overflow.

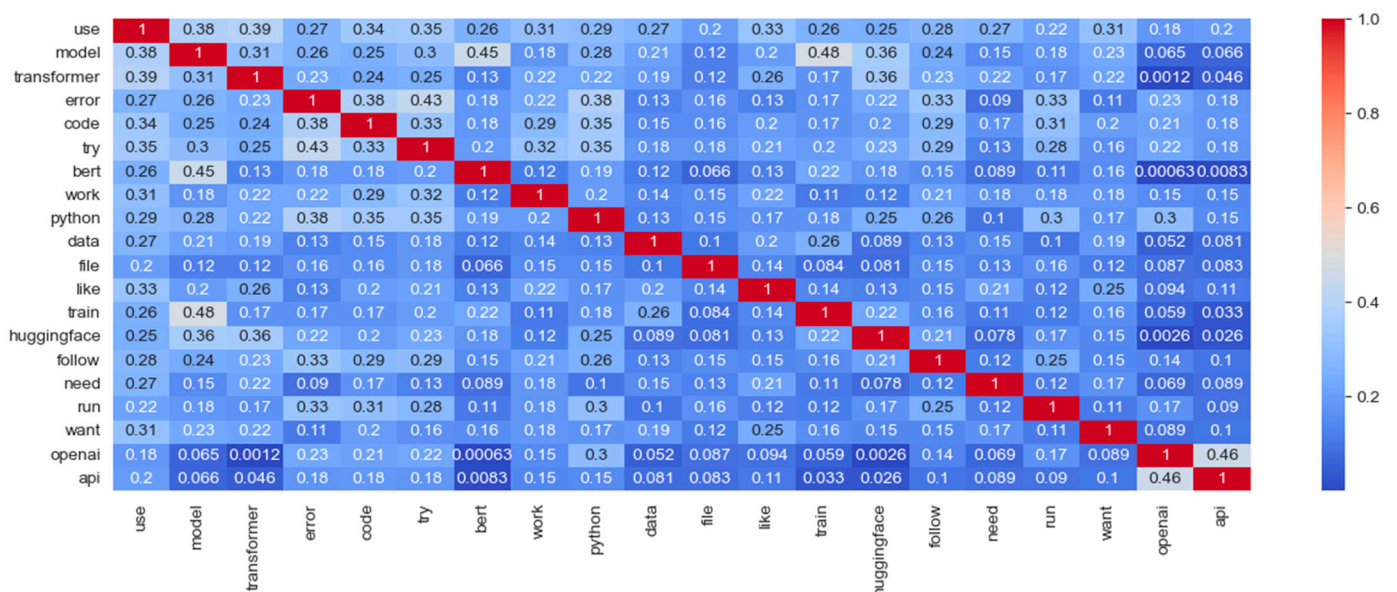


Figure 4. Cosine similarity between top TF-IDF terms.

3.4. Topic Detection and Analysis Using LDA

Latent Dirichlet Allocation (LDA) was used to find document topics based on the words in it; this research covered the discussions related to LLMs on Stack Overflow. The LDA model postulates that documents contain a blend of topics, with each topic embodying a conglomerate of words, iteratively assigning words to topics, and refining these via successive iterations. A critical step in applying LDA involves identifying the optimal number of topics (K) with the Coherence Score, which examines the word similarity within the topics and serves as an essential metric. Despite six topics providing the highest C_V score, this study strategically selected five topics, understanding that topic selection is not solely guided by coherence scores but also by the researcher's expertise (see Table 3).

Table 3. C_V coherence scores.

Number of Topics	Coherence Score	Number of Topics	Coherence Score
3	0.464357	12	0.466664
4	0.470718	13	0.465737
5	0.477660	14	0.455902
6	0.493817	15	0.452959
7	0.475287	16	0.448704
8	0.460076	17	0.458541
9	0.446352	18	0.454590
10	0.474169	19	0.432896
11	0.451729	20	0.433316

In the execution of the LDA model using the Gensim library, default hyperparameters were utilized: alpha and eta adopted a prior symmetric Dirichlet distribution, with configurations set for a single pass and multiple iterations through the corpus (alpha = symmetric; eta = none; passes = 1; and iterations = 50), ensuring harmonious equilibrium between computational efficiency and model performance.

As shown in Table 4, a total of five topics were identified: "OpenAI Ecosystem and Challenges", "LLM Training with Frameworks", "APIs, File Handling and App Development", "Programming Construction and LLM Integration", and "Data Processing and LLM Functionalities".

Table 4. The key topics about LLMs on Stack Overflow.

Section	Topic and Description	Related Terms
1	OpenAI Ecosystem and Challenges: Discussions revolve around the intricacies of deploying LLMs using OpenAI's API, emphasizing common challenges, Python-related solutions, and runtime nuances.	openai, python, code, error, run, api, gpt, follow, time, version
2	LLM Training with Frameworks: Focuses on prevalent models like BERT and the extensive use of libraries like Hugging Face for training LLMs within Python environments.	model, huggingface, train, transformer, bert, code, text, dataset, python, data
3	APIs, File Handling and App Development: Emphasizes the significance of APIs in LLMs, error resolutions, and the intricacies of developing applications that seamlessly integrate these models.	file, api, error, code, response, create, app, data, find, send
4	Programming Constructs and LLM Integration: Conversations lean towards object-oriented programming, leveraging transformers in coding paradigms, TypeScript applications, and XML integrations.	class, object, transformer, method, error, type, value, typescript, xml, code
5	Data Processing and LLM Functionalities: Highlights the nuances of data manipulations, coding strategies, and functionalities tailor-made for optimizing LLM operations.	transformer, data, function, code, type, example, column, way, value, pipeline

First, in the realm of the OpenAI Ecosystem and Challenges, the discussions predominantly center around the deployment of LLMs using OpenAI's API. The developers frequently confront Python-specific errors, runtime issues, and the challenges related to versioning to seamlessly integrate OpenAI's LLMs into diverse applications.

Second, when it comes to LLM Training with Frameworks, there is a palpable emphasis on the complexities of training these models, particularly leveraging renowned models like BERT and indispensable libraries like Hugging Face. The discourse is rich with practical facets of managing datasets, crafting code, and efficiently handling text data.

Third, the theme of APIs, File Handling, and App Development underscores the growing integration of LLMs with everyday applications. The pivotal role of APIs is clear, with developers navigating a maze of challenges, especially in the arenas of data management and the construction of LLM-centric applications.

Fourth, in the domain of Programming Constructs and LLM Integration, the narrative shifts to the confluence of object-oriented programming and LLMs. A discernible enthusiasm exists among the developers regarding how transformers dovetail into diverse coding paradigms, with a spotlight on modern web development frameworks and XML integrations.

Lastly, with Data Processing and LLM Functionalities, the crux lies in the precision of data processing tailored for LLM effectiveness. Conversations are rife with data transformation strategies, a deep dive into LLM-specific functionalities, and the pursuit of operational excellence.

t-Distributed Stochastic Neighbor Embedding (t-SNE), an advanced technique for visualizing high-dimensional data, was used to facilitate the analysis by positioning the topic distribution from the LDA model and exploring the relationships between the identified topics. Figure 5 shows that the five topics are clearly classified into distinct clusters.

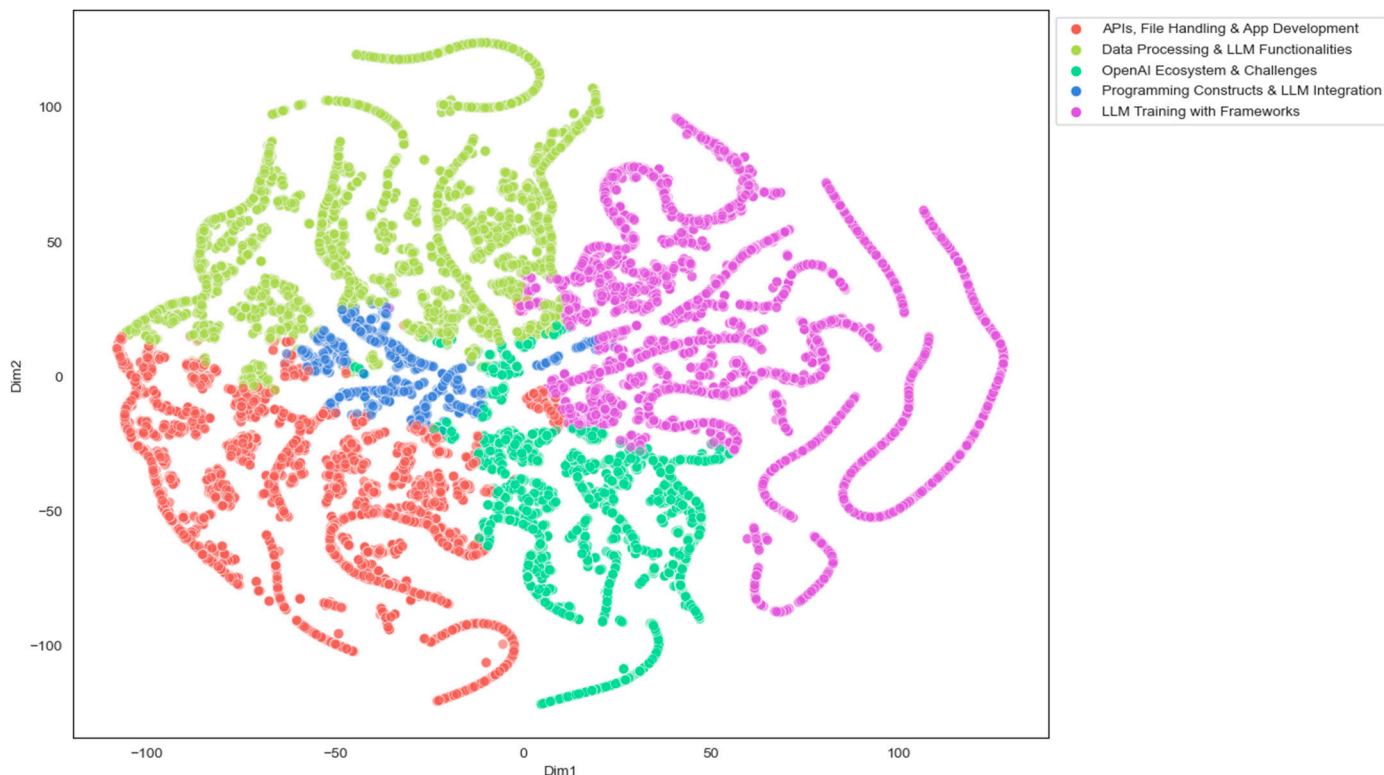


Figure 5. t-SNE representation of the topics.

3.5. Network Analysis

As shown in Figure 6, network analysis was conducted to analyze the interrelationship of the keywords derived from the LDA model. In this network, individual nodes represent

keywords, and edges represent the collective appearance of these keywords within a particular subject. The frequency of their simultaneous occurrence determines the weight or intensity of each connection edge.

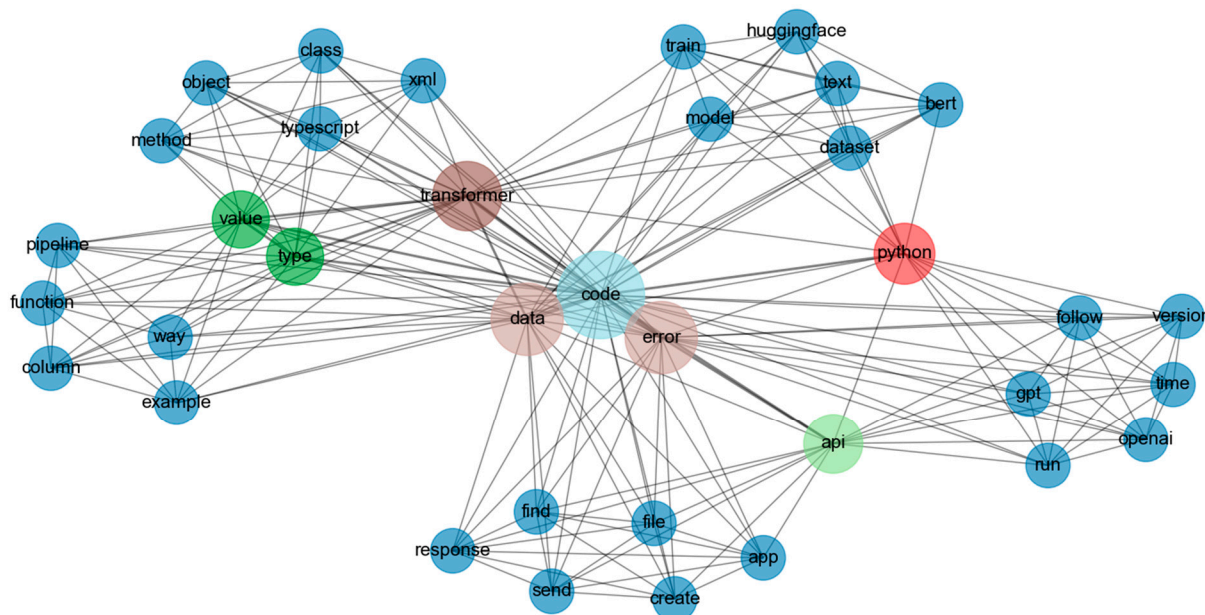


Figure 6. The result of network analysis.

The analysis confirmed that nodes, such as “transformer”, “code”, “python”, “error”, “data”, “value”, “type”, and “api”, are highly central and that these pivotal keywords play an important role in various discussions of other topics. A particularly striking feature is the strong link between “code”, “transformer”, and “python”, which has led to frequent discussions exploring the complexity associated with various transformer models. Additionally, a cluster that includes “python”, “api”, and “openai” also indicates a common narrative, concentrating on the processes and methodologies of implementing OpenAI APIs using Python. In contrast, some keywords that appear to be on the periphery with fewer connections may represent niche or specialized discussions that are less interwoven with the central discourse.

4. Conclusions

4.1. Findings and Implications

In the ever-evolving landscape of LLMs, understanding the nuances of developers’ interactions and preferences is paramount. The discourse surrounding LLMs, particularly on platforms like Stack Overflow, provides insights into the prevailing thoughts and trends within the developer community. This research has unearthed several pivotal insights into this discourse.

Firstly, this analysis underscores the significant roles of “Transformer” and “Hugging Face” within LLM discourse, reflecting prevalent developer preferences and tool use in this rapidly advancing field. The continuous development of LLM technologies necessitates consideration for the advancements of “Transformer” and “Hugging Face” and sustained engagement with expert discourse. Enhancements and development in modules, especially Hugging Face, which facilitates the use of the “Transformer Architecture”, warrant attention. With the aim of democratizing machine learning, Hugging Face alleviates computational cost barriers and simplifies natural language processing, offering customizable pre-trained SOTA models for various tasks. Consequently, the evolution of such architectures and the proliferation of open-source resources may exert a substantial influence on LLM’s technological and popular development.

Secondly, this study presented the key terms and topics of LLM expert groups to improve their expertise by looking at the trends in LLM discussions within the community. In particular, the results of the topic analysis show that “LLM Training with Frames” has more influence and is discussed more than the other topics. This means that the keywords model, Hugging Face, train, transformer, BERT, code, text, dataset, Python, data, etc., that make up “LLM Training with Frames” are the most important issues of discussion by LLM experts, and above all, can emphasize the importance of learning through the framework. Eventually, considering the complexity and rapid changes in the OpenAI ecosystem, it is necessary to learn about LLMs and discuss ecosystem expansion and development through API integration.

Finally, as you can see from the network analysis results, it was confirmed that a structural relationship between the topics and terms centered on transformers, code, and Python appeared. Ultimately, these results show that the community of LLM experts builds ecosystems and works on structural network communication. In the end, the LLM also shows that efforts to expand the ecosystem based on global networks and cooperation are of paramount importance and not the domain of specific experts or institutions. Therefore, other than the open sourcing of technology and data in the future, it is necessary to create a foundation for global cooperation by experts to further developmental discourse and strengthen business utilization.

These sequential insights have significant implications. For developers, they offer a roadmap to understand the prevailing trends, aiding their learning of LLM advancements. For researchers, the findings from this study present valuable directions for future research, especially in areas with pronounced interest or noticeable gaps. Developers can utilize identified trends to inform technology adoption, while researchers may explore the causes behind the prominence of certain technologies in LLM discourse. Tool creators can align tool development with prevalent community topics and challenges, ensuring relevance and utility for LLM developers and practitioners.

Moreover, the insights derived could potentially steer the evolution and adoption of LLM technologies across varied applications by highlighting the focal points and challenges within the developer community. This, in turn, could foster a more informed and collaborative approach towards addressing prevalent issues and innovating solutions in the LLM domain, thereby bridging the gap between theoretical advancements and practical implementations. The insights derived could potentially guide the development and adoption of LLM technologies in various applications. By highlighting the key areas of focus and challenges within the developer community, this can promote a more informed and collaborative approach to addressing the existing LLM issues and innovative solutions. As a result, it can bridge the gap between theoretical advancements and practical implementations.

In particular, many companies continue to enhance their efficiency by incorporating LLM-based technologies into management systems and processes. Accordingly, LLM is an object that must be learned by many people in the entrepreneurial and business sectors. In this respect, the results of this study will help them understand LLM’s technology trends and suggest learning directions for business utilization. As collaboration and communication with LLM experts should continue to take place within an open ecosystem, a collaborative learning program should be considered.

4.2. Limitations and Future Plans

Although this study offers a comprehensive and insightful exploration of the LLM discourse on Stack Overflow, it possesses certain limitations that must be acknowledged. Firstly, despite the notable depth provided by focusing the data analysis on Stack Overflow, it is imperative to acknowledge the potential biases and limitations, given that the dialogue therein may not wholly represent the broader, multifaceted discussions and perspectives on LLM prevalent across various forums and communities. In the future, researchers must cautiously interpret these insights as a part of a larger, global developer dialogue

and potentially extend the scope by incorporating data from diverse platforms for a more comprehensive and inclusive view of global LLM discourse.

Secondly, a notable temporal limitation is present in our study. The cut-off of early June 2023 precludes the most recent three months, during which there has been a discernible uptick in discussions and releases pertaining to open-source LLMs. Consequently, this study might not encapsulate the nuances, challenges, and innovations introduced by these nascent open-source LLM developments. These omissions underscore potential research gaps that should be addressed in the future to ensure the timely and contemporaneous portrayal of ever-evolving LLMs.

Lastly, other researchers could explore the use of tools such as the transformer-based BERTopic for a more nuanced examination of the discourse surrounding LLMs. BERTopic, with its ability to comprehend the semantic context of words and phrases, could potentially provide richer and more meaningful topic representations than traditional methods like TF-IDF and LDA can. Further analyzing these extracted topics using models like GPT or LLaMA could provide deeper insights into the underlying themes and enhance our understanding of the text data.

Author Contributions: Conceptualization, J.S. and B.K.; methodology, J.S.; software, J.S.; validation, J.S.; formal analysis, J.S.; investigation, J.S.; resources, J.S.; data curation, J.S.; writing—original draft preparation, J.S. and B.K.; writing—review and editing, B.K.; visualization, B.K.; supervision, B.K.; project administration, B.K.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Teubner, T.; Flath, C.M.; Weinhardt, C.; van der Aalst, W.; Hinz, O. Welcome to the era of chatgpt et al. the prospects of large language models. *Bus. Inf. Syst. Eng.* **2023**, *65*, 95–101. [\[CrossRef\]](#)
2. De Angelis, L.; Baglivo, F.; Arzilli, G.; Privitera, G.P.; Ferragina, P.; Tozzi, A.E.; Rizzo, C. ChatGPT and the rise of large language models: The new AI-driven infodemic threat in public health. *Front. Public Health* **2023**, *11*, 1166120. [\[CrossRef\]](#)
3. Roumeliotis, K.I.; Tselikas, N.D. ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet* **2023**, *15*, 192. [\[CrossRef\]](#)
4. Thakur, N. Monkeypox2022tweets: A large-scale twitter dataset on the 2022 monkeypox outbreak, findings from analysis of tweets, and open research questions. *Infect. Dis. Rep.* **2022**, *14*, 855–883. [\[CrossRef\]](#)
5. Singh, J. An efficient deep neural network model for music classification. *Int. J. Web Sci.* **2022**, *3*, 236–248. [\[CrossRef\]](#)
6. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S. Palm: Scaling language modeling with pathways. *arXiv* **2022**, arXiv:2204.02311.
7. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
8. Jo, A. The promise and peril of generative AI. *Nature* **2023**, *614*, 214–216.
9. Vaithilingam, P.; Zhang, T.; Glassman, E.L. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In Proceedings of the Chi Conference on Human Factors in Computing Systems Extended Abstracts, New Orleans, LA, USA, 30 April–5 May 2022; pp. 1–7.
10. Thakur, S.; Ahmad, B.; Fan, Z.; Pearce, H.; Tan, B.; Karri, R.; Dolan-Gavitt, B.; Garg, S. Benchmarking Large Language Models for Automated Verilog RTL Code Generation. In Proceedings of the 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 17–19 April 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
11. Li, H. Language models: Past, present, and future. *Commun. ACM* **2022**, *65*, 56–63. [\[CrossRef\]](#)
12. Hussain, Z.; Nurminen, J.K.; Mikkonen, T.; Kowiel, M. Combining Rule-Based System and Machine Learning to Classify Semi-natural Language Data. In Proceedings of the SAI Intelligent Systems Conference, Amsterdam, The Netherlands, 1–2 September 2022; Springer: Berlin/Heidelberg, Germany; pp. 424–441.
13. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [\[CrossRef\]](#)

14. Bengio, Y.; Ducharme, R.; Vincent, P. A neural probabilistic language model. In Proceedings of the Neural Information Processing Systems (NIPS2000), Denver, CO, USA, 1 January 2000.
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Neural Information Processing Systems (NIPS2017), Long Beach, CA, USA, 4–9 December 2017.
17. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
18. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process Syst.* **2020**, *33*, 1877–1901.
19. Peruma, A.; Simmons, S.; AlOmar, E.A.; Newman, C.D.; Mkaouer, M.W.; Ouni, A. How do i refactor this? An empirical study on refactoring trends and topics in Stack Overflow. *Empir. Softw. Eng.* **2022**, *27*, 11. [[CrossRef](#)]
20. Han, J.; Shihab, E.; Wan, Z.; Deng, S.; Xia, X. What do programmers discuss about deep learning frameworks. *Empir. Softw. Eng.* **2020**, *25*, 2694–2747.
21. Li, J.; Tang, T.; Zhao, W.X.; Nie, J.-Y.; Wen, J.-R. Pretrained language models for text generation: A survey. *arXiv* **2022**, arXiv:2201.05273.
22. Barua, A.; Thomas, S.W.; Hassan, A.E. What are developers talking about? an analysis of topics and trends in stack overflow. *Empir. Softw. Eng.* **2014**, *19*, 619–654. [[CrossRef](#)]
23. Ford, D.; Smith, J.; Guo, P.J.; Parnin, C. Paradise unplugged: Identifying barriers for female participation on stack overflow. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Seattle, WA, USA, 13–18 November 2016; pp. 846–857.
24. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
25. Asaduzzaman, M.; Mashiyat, A.S.; Roy, C.K.; Schneider, K.A. Answering questions about unanswered questions of stack overflow. In Proceedings of the 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, USA, 18–19 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 97–100.
26. Yazdani, R.; West, R.; Dillenbourg, P. Keeping up with the trends: Analyzing the dynamics of online learning and hiring platforms in the software programming domain. *Int. J. Artif. Intell. Educ.* **2021**, *31*, 896–939. [[CrossRef](#)]
27. Omondigbe, O.P.; Licorish, S.A.; MacDonell, S.G. Features that predict the acceptability of java and javascript answers on stack overflow. In Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering, Copenhagen, Denmark, 14–17 April 2019; pp. 101–110.
28. Rodríguez, L.J.; Wang, X.; Kuang, J. Insights on apache spark usage by mining stack overflow questions. In Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, USA, 2–7 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 219–223.
29. Ithipathachai, V.; Azizi, M. Are tags' it? Analysis of the impact of tags on StackOverflow questions. In Proceedings of the 37th ACM/SIGAPP. Symposium on Applied Computing, Virtual Event, 25–29 April 2022; pp. 1483–1490.
30. Creative Commons. Available online: <https://creativecommons.org/licenses/by-sa/4.0/deed.en> (accessed on 20 August 2023).
31. Stack Exchange Data Dump. Available online: <https://archive.org/details/stackexchange> (accessed on 30 July 2023).
32. Zhu, W.; Zhang, H.; Hassan, A.E.; Godfrey, M.W. An empirical study of question discussions on Stack Overflow. *Empir. Softw. Eng.* **2022**, *27*, 148. [[CrossRef](#)]
33. Linares-Vásquez, M.; Bavota, G.; Di Penta, M.; Oliveto, R.; Poshvanyk, D. How do api changes trigger stack overflow discussions? A study on the android sdk. In Proceedings of the 22nd International Conference on Program Comprehension, Hyderabad, India, 2–3 June 2014; pp. 83–94.
34. Bird, S. NLTK: The natural language toolkit. In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, 17–18 July 2006; pp. 69–72.
35. Rehurek, R.; Sojka, P. *Gensim—Python Framework for Vector Space Modelling*; NLP Centre, Faculty of Informatics, Masaryk University: Brno, Czech Republic, 2011.
36. Industrial-Strength Natural Language Processing. Available online: <https://spacy.io> (accessed on 30 July 2023).
37. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21. [[CrossRef](#)]
38. Xiao, K.; Qian, Z.; Qin, B. A survey of data representation for multi-modality event detection and evolution. *Appl. Sci.* **2022**, *12*, 2204. [[CrossRef](#)]
39. Fan, H.; Du, W.; Dahou, A.; Ewees, A.A.; Yousri, D.; Elaziz, M.A.; Elsheikh, A.H.; Abualigah, L.; Al-qaness, M.A.A. Social media toxicity classification using deep learning: Real-world application UK Brexit. *Electronics* **2021**, *10*, 1332. [[CrossRef](#)]
40. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
41. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
42. Houslyby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 2790–2799.
43. Hagberg, A.; Swart, P.; Schult, D. *Exploring Network Structure, Dynamics, and Function Using Networkx*; Los Alamos National Lab. (LANL): Los Alamos, NM, USA, 2008.

44. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
45. LangChain. Available online: https://python.langchain.com/docs/get_started/introduction (accessed on 15 August 2023).
46. Topsakal, O.; Akinci, T.C. Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. In Proceedings of the International Conference on Applied Engineering and Natural Sciences, Konya, Turkey, 10–12 July 2023; Volume 1, pp. 1050–1056.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.