EXPLORING ARCHITECTURAL KNOWLEDGE IN BLOGS

Kirsten Gericke
s4041976

Supervisor: Dr Mohamed Soliman

A thesis presented for the degree of
Bachelor of Computing Science

Computing Science
University of Groningen
August 2022

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Mohammed Soliman his invaluable feedback, knowledge and expertise, and for guiding me through this process.

ABSTRACT

Software engineers need access to architectural knowledge concepts to make relevant architectural design decisions and fulfil the quality attributes of a system. Out of the many sources of architectural knowledge that exist, blog web pages have been found to contain the most architectural knowledge. The goal of this project is to classify types and topics of architectural blogs, and architectural knowledge concepts discussed in architectural blogs. This will help software engineers access relevant architectural knowledge and aid the future development of abstraction tools. Both inductive (Grounded Theory) and deductive (Qualitative Content Analysis) analysis techniques will be used to achieve this goal.

LIST OF TABLES

List of Tables

LIST OF FIGURES

List of Figures

Contents

1 Introduction

Software architectural design decisions (ADDs) play a crucial role when developing a software system [1]. Making the right ADDs is a challenging task, which requires knowledge and expertise mostly acquired through experience. Without the required architectural knowledge (AK), software engineers might make uncertain and risky assumptions about the ADDs, and therefore need to be able to search and locate it [11]. However, software developers face challenges to find AK, because software developers do not commonly document design decisions.

AK concepts are conceptual elements that describe and characterise AK [11]. Some examples of AK concepts are technology features, benefits and drawbacks, and requirements and constraints. These concepts can be captured and sourced from technology documentation, issue tracking systems, and developer communities (eg. Stackoverflow) [14], along with countless others. This research project focuses on sourcing AK from blogs on the web.

Software blogs are articles written by software developers to share their technical experiences. Architectural blogs discuss the architectural design of software systems, architectural solutions (eg. patterns) and design decisions. A wide variety of blogs exist, such as community blogs (eg. DZone), personal blogs (eg. Martin Fowler), blogs published by technology providers (eg. Microsoft) or consulting companies, etc. [14]. These articles are useful for sharing AK with software developers, however little is known about the types of architectural blogs, their sources and contents.

There are a number of studies classifying AK concepts, building repositories to store AK, and searching for AK in various sources including web pages [14], developer communities [12] and issue tracking systems [11]. This existing research contains several shortcomings rooted in not recognizing the ability of blogs in helping software engineers and the prevalence of AK concepts in blogs. It is known that architectural blogs contain the most AK out of all web based sources [14], therefore focusing on blogs rather than other sources will provide the most useful and generalisable results. This is hence why blogs have been selected for this project rather than other sources of AK. Additionally, blogs are an heterogeneous source of AK, such that capturing this knowledge is only possible if systematic processes are defined specifically for blogs. This project addresses these shortcomings by capturing and organising relevant and up-to-date AK from experienced software developers in architectural blogs.

The goal of this project is to identify and classify architectural blog types, topics and AK concepts using Grounded Theory and Qualitative Content Analysis.

The Research Question can be summarised as follows: *What are the types, topics and architectural knowledge concepts discussed in architectural blogs?*. This will be answered in multiple steps, first focusing on types, then topics, and finally AK concepts.

Exploring AK in various sources will positively impact the field of software architecture such that the ADDs made would better fulfill required quality attributes (eg. performance, reliability, security, scalability, etc. [1]) and reduce risky assumptions about selection, creation and evaluation of a software architecture. Focusing specifically on blogs on web pages aims to perform this task with maximum generalisability based on scientific research proving blogs have the largest amount of available AK data. This systematic approach is designed to be repeatable and aid the development of future abstraction tools to easily allow software engineers access to this knowledge.

This project lead to the following contributions:

- 926 architectural blogs classified into types
- 257 architectural blogs classified into topics
- 1662 AK concepts annotated from 35 architectural blogs
- The following lists (including definitions, examples, frequency and common indicators):
  - A list of 7 types of architectural blogs
  - A list of 5 topics of architectural blogs
  - A list of 13 AK concepts discussed in architectural blogs
- Analysis of relationships between AK concepts and types of blogs, and AK concepts and topics of blogs

These contributions will provide software engineers more specific criteria when searching for AK and will save time by avoiding searching the wrong blog type or topic for the required AK.

The supervisor of this project has been involved in previous studies involving AK [10, 11, 12, 13, 14]. The data set for this project was retrieved from Soliman et al. [14]. The AK concepts used in this project were retrieved from Soliman et al. [11, 12, 14].

The research project is structured as follows: Section 2 provides background on existing studies related to AK concepts in web based sources. Section 3 explains the methods and processes implemented to answer the research question of this project, separated into three steps. Section 4 presents the results from each of these three steps, followed by a discussion in Section 5 and finally the conclusion in Section 6.

2 Background

2.1 Architectural Knowledge

This project will use AK concepts gathered from three studies performed by Soliman et al. [11, 12, 14]. Please refer to Table 4 which shows which AK concepts were retrieved from each study, and Table 19 which contains the definitions of the AK concepts used in this project. Please also refer to Section 3.1 for an explanation on the creation of the dataset from Soliman et al. [14].

Previous research in the field of AK has explored design decisions [7, 10] and the concept of an AK repository system [8].

This project assumes the definitions by Jansen and Bosch [7] that describe ADDs as first-class entities and define software architecture as a set of ADDs in order to reduce knowledge vaporisation.

Kruchten et al. [8] suggests that an explicit representation of AK as a repository would be helpful for building and evolving quality systems, rather than AK remaining solely in the heads of software architects. They identify three levels of knowledge that may also be applied to architectural knowledge: Tacit (mostly in the head of people), Documented (there is some trace somewhere), and Formalised (not only documented, but organised in a systematic way). Moving AK from the tacit level to the documented and formalised level will allow future evolutionary capabilities of a system to be better assessed [8]. Classifying and annotating AK concepts in blogs does exactly that: organise knowledge that would otherwise be difficult or impossible to access. This idea further emphasises the reasoning and importance of exploring AK knowledge in blogs.

Soliman et al. [10] propose two possible reasoning types as to why software architects make ADDs: deductive/problem driven or inductive/solution driven. It is important to understand these different reasoning methods in order to identify the AK concepts and relationships in blogs. They suggest ADDs are only reusable when justified based on architectural concerns, and not when based on business or social aspects. This suggests not all AK in blogs is reusable, and hence results would need to be filtered in future work when tools or repositories are developed to ensure only reusable AK is shared amongst software engineers.

Lack of documentation could be due to the architect being unaware of the ADD, the company tactically excluding it, or the architect excluding it for no valid reason (eg. forgetting or overlooking its importance). Regardless of the reason, this lack of documentation leads to vaporisation of decisions and rationale, more specifically knowledge that influences design decisions and reasoning about if and why one design decision is more important than another [8]. It is this important information that this project aims to capture in blogs, to therefore reduce its potential vaporisation.

2.2 Web Based AK Sources

Researchers have explored and captured AK concepts in multiple different sources, such as developer communities (e.g. Stack Overflow [3, 12, 13]), Google search results [14], technology documentation [6], mailing lists [5] and issue tracking systems [2, 11]. However, these studies do not explore or capture AK in the most useful source: blogs.

It is important to distinguish sources that discuss coding problems or development issues (eg. Bug fixing, testing) rather than architectural issues or concepts. An example is Stack Overflow, which typically focuses on lower level implementation details that are irrelevant to software architects. However it was shown by Soliman et al. [12] that Stack Overflow does in fact also contain useful AK through developing an AK concept ontology. They defined both `Simple` and `Composite Significant AK Concept classes` which are used in this project to categorise AK concepts in blogs. Stack Overflow is more structured than architectural blogs, however they have clear similarities in their content and authors, and are both heterogeneous web based AK sources. Therefore it is appropriate to draw a basis of AK for blogs from this research.

Soliman et al. [11] discusses how AK concepts are textually represented in issue tracking systems using keywords, adjectives, quality measurement, and other textual variants. For example, benefits and drawbacks of solutions are expressed explicitly using specific keywords ("advantages", "limitations") and adjectives ("good", "ugly", "fragile", "general"). This approach of recognizing textual keywords during AK concept classification can be used in this project to determine common indicators of blog types, topics and AK concepts. A method such as this helps create a structured and repeatable research process.

Soliman et al. [14] provides the foundation for this research project. They collected types of web based AK sources by automatically clustering URLs of web pages using a clustering algorithm, resulting in 31 clusters. These clusters were refined down to 9 final categories of AK sources by determining the dominant AK source in each cluster by manually classifying the web pages. These 9 categories are shown in Table 1. Descriptive statistics and Pearson's chi-squared correlation test were used to determine which AK concepts exist in each source. The results identify blogs as having the highest relevance, ranking, and amount of AK compared to other web based sources, and hence explains why it is the most required area for further research.

Table 1: AK sources on the web and their percentage of the total amount of sources retrieved by Soliman et al. [14].

AK source	**Percentage**
Blogs and tutorials	39%
Technology vendor documentations	23%
Scientific contents	13%
Forums	7%
Technical books and white papers	4.50%
Source code repositories	4.50%
Knowledge repositories	4%
Presentations and videos	2.70%
Others (eg. tools, patents)	2.30%

2.3 Background Conclusion

There are a number of studies classifying AK concepts, building repositories to store AK, and searching for AK in various sources. Much of this information is applicable and useful for this research project and future work, however no exploratory research has been done on the contents of AK in blogs. With the exception of Soliman et al. [14], these studies share a common shortcoming: they overlook the importance of blogs in helping software engineers and the prevalence of AK concepts in blogs. This is done through either focusing on a different type of source or ignoring the fact that blogs contain the most AK. This project addresses these shortcomings.

2.4 Literature Collection

The state of the art research began by analysing the impressive amount of related research articles published by Dr. Mohamed Soliman [10, 11, 12, 13, 14], the supervisor of this research project, whom provided multiple literature sources covering related research studies and the concepts, explanations and methods used in the field of Software Architecture [1, 9, 15]. Digital libraries such as SmartCat, IEEE Xplore and Web of Science where used in conjunction with the general keywords *"architecture knowledge", "architecture design decisions", "software engineer*", "software architecture"* and more specific keywords *"blogs", "AK concepts", "topic modeling"* and *"grounded theory"*.

3 Methodology

This project consisted of 3 steps, each answering a Research Question:

- **RQ1: What are the types of architectural blogs?** This step was executed using Grounded Theory, a research method used in qualitative studies in order to develop theories from data [15]. Due to its inductive nature, the types of architectural blogs were not required to be known before analysis took place.
- **RQ2: What topics are discussed in architectural blogs?** Similarly to Step 1, Step 2 was executed using Grounded Theory. A subset of blogs were further categorised into topics.
- **RQ3: What AK concepts are discussed in architectural blogs?** A subset of blogs was uploaded to Atlas.ti in which Qualitative Content Analysis was used to annotate AK concepts. The AK concepts used during this step were gathered from existing literature, hence a deductive approach was taken in contrast to Step 1 and 2.

Subsection 3.1 provides background on the dataset and the structure of the study performed in Soliman et al. [14]. The following 3 subsections describe the methodology of each of the 3 steps executed in this project.

3.1 Data Collection

The dataset was collected from an empirical study by Soliman et al. [14], in which 53 software engineers used Google to make design decisions using the Attribute-Driven-Design method proposed by Kazman et al. [4]. This process contains 3 ADD steps (Identify design concepts, Select design concepts and Instantiate architecture elements). The participants solved six architectural design searching tasks. For each searching task, the participants performed one of the three ADD steps through executing multiple queries in Google, and assessing the resulted web pages regarding two aspects: 1) The relevance of each web page to the searching task, and 2) The AK concepts which exist in each web page. The submitted results (relevance and AK concepts) were stored in a database through a Google Chrome plugin (provided online).

Soliman et al. [14] collected 2623 unique web pages, of which 39% (1023) were categorized as "blogs and tutorials". 945 of these were blogs. The URLs of the 945 blogs were provided to me by my supervisor in a file titled `Blogs starting set.xls`. The relevance rankings for these blogs were collected from the file `usersUrls.csv`, which contains the participants' usernames, taskname, searchqueryid, URL, and relevance rankings (on a scale from 1 to 5). The Github repository for Soliman et al. [14] can be viewed at https://github.com/m-a-m-s/ICSA2021.

The participants of Soliman et al. [14] could choose a degree of relevance from a five-level Likert scale with the following definitions:

1. *"No Relevance (N): The web page has nothing to do with the task. It has no relevant information.*
2. *Low Relevance (L): The web page contains information, which is only remotely relevant to solving the given task, but might help for refining the search.*
3. *Medium Relevance (M): The web page addresses a different problem to that of to the task at hand, but it provides some relevant information to the task, which could be an answer to the searching goal. Nevertheless, the provided information does not match specifically the task's requirements.*
4. *High Relevance (H): The web page addresses a similar problem to that of the task and contains useful information. The web page provides an answer to the searching goal, and helps with fulfilling one requirement of the task.*
5. *Very High Relevance (VH): The web page discusses a similar problem to that of the task and contains useful information. The web page provides an answer to the searching goal, and helps with fulfilling more than one requirement of the task."*

Note that in the dataset used in this project, these values were represented numerically from 1 to 5, rather than alphabetically from N to VH.

In order to perform the second assessment of identifying AK concepts, the participants were provided the following five of AK concepts: Solution description, Solution alternatives, Solutions benefits, Solutions drawbacks, Made design decisions, Others. This list of AK concepts needed to be limited due to the nature of the study in order to limit the complexity of results considering the high number of participants and tasks. This list however is considered too limited and simple for this project. The list of AK concepts for this project was therefore gathered from Soliman et al. [11, 12], which the exception of the AK concept "Solution alternatives" taken from Soliman et al. [14]. The process of refining the list of AK concepts used in this project is explained in Section 3.4.

3.2 Methodology Step 1: Types

Grounded Theory (GT) refers to a method of inductively generating theory from data, rather than validating a theory. GT studies can be performed on unstructured text (such as interview transcripts, documents and field notes), structured text, diagrams and images [15]. Out of the three main streams of GT (Glaser's GT (classic or Glaserian GT), Strauss and Corbin's GT (Straussian GT), and Charmaz's constructivist GT [15]), classic CT is used in this project. GT has multiple core features proposed by Stol et al. [15]. These features (shown in bold) along with explanations on how each feature was executed in this project is as follows:

- **Limit exposure to literature:** in order to promote open-mindedness and limit thinking in terms of established concepts, no prior background research into classifying blog types was conducted.
- **Treat everything as data:** when examining a blog, I would take the entire page into account, including the headers and footers, menu items, title, subtitle, author's name, URL, images, logos, etc.
- **Immediate and continuous data analysis:** I simultaneously performed data collection and analysis through opening a blog page, examining the features, deciding which category is applicable, and noting down the type and reason next to the URL in the spreadsheet.
- **Theoretical sampling:** in order to identify further data sources based on gaps in the emerging theory and further explore unsaturated concepts, I expanded my dataset from an initial subset of 300 blogs, to the entire set 945 blogs.
- **Theoretical sensitivity:** I conceptualized and established relationships between concepts by determining the features of blogs (eg. title, author, keywords, etc.) that differ and can be grouped (eg. one author vs multiple authors) that contribute to classifying the blog as a certain type.
- **Coding:** I used inductive and abductive logic through inferring the blog type categories from the data and each blog's properties, rather than using a preconceived coding scheme or inferring categories from the hypothesis. I created a column in the spreadsheet describing why a blog type was chosen for each URL, and all properties of that blog that contributed to that classification (eg. where I found the information on the page, number of authors, quotes from the page, etc.).
- **Memoing:** In order to elaborate categories, describe preliminary properties, describe relationships between categories, and identify gaps in the data, I created memos in the form of a Google Document in which I created lists of common patterns (wording, properties, locations of important information) that I would recognize, along with tables containing definitions, examples, sub-categories, and attributes. These memos, tables and definitions were constantly being altered when new information was found.
- **Constant comparison:** The more blogs I classified, the more I began to recognize the important and unimportant properties that contributed to a blog type, such that near the end of Step 1 I knew exactly what to look for to determine what the blog type was (refer to Table 6 for these attributes). Since my understanding of differing blog types became clearer as I analysed more blogs, I often revisited previously classified blogs, for example to see whether the value of a new property aligned with my new conceptual understanding of that blog type. Considering I reiterated over the dataset multiple

times (the changes made during the four iterations are described in item number four in the list below), the names, definitions and understandings of each type category evolved over time, with each iteration being influenced by the data, my memos and my codes.

- **Memo sorting:** I oscillated between my codes in the spreadsheet and the emerging theory outline from my Google Document and memos, often updating and improving the definitions and important attributes of each type.
- **Cohesive theory:** I attempted to develop a cohesive theory of blog type categories and move beyond superficial categories by considering multiple properties and exploring the website as a whole, for example viewing other pages (eg. `About` page) or blog posts on the same website (eg. to determine how many authors post on the website).
- **Theoretical saturation:** I stopped collecting data once theoretical saturation was reached such that all 945 blogs in the dataset were classified into a blog type.

A sequential explanation of the tasks performed during Step 1 is as follows:

1. The files `Blogs_starting_set.xlsx` and `usersUrls.csv` were combined. Considering multiple participants from Soliman et al. [14] performed the same task, there existed web pages that were assigned a relevance ranking by multiple participants, and therefore appeared multiple times in the dataset, and therefore required their relevance rankings to be averaged. This was calculated using the Excel formula `AVERAGEIF`. The frequency of each relevance ranking for the set of 945 blogs, excluding those not assigned a relevance, can be seen in Figure 1.

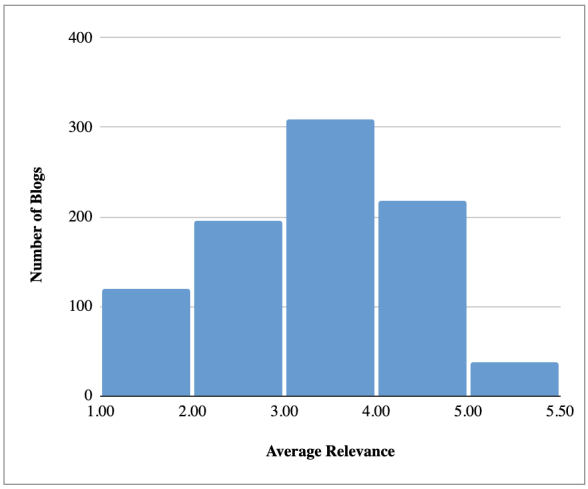


Figure 1: Histogram of Average Relevance Rankings collected from [14], for set of initial 945 blogs.

2. 18 URLs were removed such that the web page redirected to a URL already existing in the spreadsheet and therefore was considered a duplicate. For example, the URL was identical to another, except ended in a hashtag. Removing these duplicates resulted in a total of 926 URLs.
3. Each type of blog is different in its structure and contents, thus in order to explore blog types and their characteristics thoroughly, I included in the spreadsheet explanations as to why a blog type was chosen for each URL, from which I was able to establish definitions, examples, sub-categories and attributes for each blog type.
4. Blog type classifications were re-iterated 4 times after discussions with my supervisor, each iterations making the following alterations:
    - **Clarifying definitions** (eg. blog posts from medium.com was changed from a "Personal Blog" to a "Community Blog" since it is possible for multiple users to post blogs on the website.)

- **Renaming categories into more accurate descriptive names** (eg. "Provider" to "Vendor", "Consulting" to "Services", "Publication" to "Magazines and Newspapers", "NA" to "Not assignable" or "Not a blog")
- **Removing categories that were very uncommon** (eg. "General blog" and "Publisher") **or had so many similarities with another category that it was not required** (eg. blogs in "Skills and Development organisation" was reclassified into either a "Tutorial" or an "Educational IT course provider")

5. The number of blogs per blog type was calculated. The relevance rankings per blog type was averaged and represented in box plots. Graphs were created from these results.

6. A Pearson correlation test was performed to measure the linear dependence between types of blogs and relevance. In R, the *cor.test()* test was used for association/correlation between paired samples. It returns both the correlation coefficient and the significance level(or p-value) of the correlation. If the p-value is $< 5\%$, then the correlation between x and y is significant. A p-value less than 0.05 is typically considered to be statistically significant, in which case the null hypothesis should be rejected.

7. The number of blogs per type was calculated for each individual task conducted by participants of Soliman et al. [14]. These values where compared for correlation and viewed in a stacked column chart.

3.3 Methodology Step 2: Topics

The same approach as Step 1 was used in Step 2. Classic Grounded Theory was used, along with a spreadsheet and the same Google Document for memoing. The explanations of how each feature (shown in bold) of GT was executed in this project during Step 2 is as follows:

- **Limit exposure to literature, Immediate and continuous data analysis, Memoing, Constant comparison, Memo sorting:** These features were performed in the same manner as Step 1. Please refer to Section 3.2 for these explanations.
- **Theoretical sampling:** The data sources identified for Step 2 were sampled methodically to maximise generalisability and accurately represent the results of Step 1, proportionate to the number of blogs in each type category and relevance ranking. Please refer to item number two in the list below for an in depth explanation and calculation of this sampling process.
- **Theoretical sensitivity:** I conceptualized and established relationships between concepts by determining the features of blogs (eg. title, keywords, etc.) that differ and can be grouped (eg. explaining one technology vs comparing two technologies vs listing multiple technologies) that contribute to classifying the blog as having a certain topic.
- **Coding:** I used inductive and abductive logic through inferring the blog topic categories from the data and each blog's properties, rather than using a preconceived coding scheme or inferring categories from the hypothesis. I created a column in the spreadsheet describing why a blog topic was chosen for each URL, (eg. quotes, keywords in the title, subtitle, or headings, summarised information the blog provided, etc.).
- **Cohesive theory:** I attempted develop a cohesive theory of blog topic categories and move beyond superficial categories by condensing multiple factors (eg. number of technology solutions, level of abstraction, does it provide explanations, advice or examples, etc.) into a single all-encompassing topic.
- **Theoretical saturation:** I stopped collecting data once theoretical saturation was reached such that the mathematically significant sample of 257 blogs in the dataset were classified into a blog topics.

A sequential explanation of the tasks performed during Step 2 is as follows:

1. Topic allocation was needed to be performed on a mathematically significant sample of the set of 926 blogs from Step 1, rather than classifying all 926 blogs. Step 1 resulted in 770 out of the 926 blogs being categorised into 7 types. I used an online sample size calculator to compute the minimum number of necessary samples to meet the desired statistical constraints, which resulted in 257. This means 257 or more measurements/blogs were needed to be classified in Step 2 to have a confidence level of 95% that the real value is within ± 5% of the measured/surveyed value.

2. I took samples from the 7 types that are proportionate to the number of blogs in that category, adding up to a sample of 257. The calculations can be seen in Table 2. Within each category, the sampling is again completed in proportion to the number of blogs per relevance ranking, resulting the the number of blogs sampled from each type category and each relevance ranking shown in Table 3. This process guaranteed that all types of blogs were included in the evaluation sample proportional to their occurrence in the overall sample.

Table 2: Calculations of the Number of Blogs sampled from each Type category for Step 2.

Type	**Number of Blogs**	**/ 770**	*** Sample Size**	**Rounded**
Personal blog	117	0.1519480519	39.05064935	39
Community blog	332	0.4311688312	110.8103896	110
Educational IT course provider	21	0.02727272727	7.009090909	7
IT Service Company	88	0.1142857143	29.37142857	30
Technology Vendor	190	0.2467532468	63.41558442	64
University	4	0.005194805195	1.335064935	1
Magazines and Newspapers	18	0.02337662338	6.007792208	6
Total	**770**	**1**	**257**	**257**

Table 3: Number of Blogs sampled from each Type category and Relevance Ranking for Step 2.

		Relevance Ranking					
Type	**Total**	**5**	**4**	**3**	**2**	**1**	**NA**
Personal blog	39	3	12	16	8	0	0
Community blog	110	6	33	47	24	0	0
Educational IT course provider	7	0	2	3	2	0	0
IT Service Company	30	2	7	12	9	0	0
Technology Vendor	64	2	17	25	20	0	0
University	1	0	0	1	0	0	0
Magazines and Newspapers	6	0	0	2	4	0	0
Total Amount of Blogs	**257**	**13**	**71**	**106**	**67**	**0**	**0**

3. When selecting the sample from the dataset of Step 1 to use in Step 2, I did not include the "Not a blog" and "Not assigned" categories for obvious reasons, as well as excluding those with a relevance ranking 1 since according to its definition, *"The web page has nothing to do with the task. It has no relevant information."*[14]. The sample for each category needed to be chosen randomly to ensure generalisable results. I used the `RANDBETWEEN(1,700)` function to generate random values for each blog, ordered them in ascending order of the randomly generated values, and selected (in ascending order) the required amount of blogs per category per relevance ranking needed. Once this set was determined, categorising the blogs by topic began.

4. Similarly to Step 1, during analysis I noted reasons as to why a blog topic was chosen in the spreadsheet, as well as established definitions and common indicators for each topic category.

5. Blog topic classifications were re-iterated three times, making the following alterations:

- **Clarifying definitions** (eg. explicitly defining the differences between "Use Case" and "How To")
- **Renaming categories into more accurate descriptive names** (eg. "List" to "List of alternative solutions")
- **Removing categories that were uncommon** (eg. "Performance Analysis", "Tips") **or had so many similarities with another category that it was not required** (eg. most blogs in "Recommendations" were reclassified into "Comparison", and most blogs in "Benefits and Drawbacks" and "Summary" were reclassified into "Solution Evaluation")
- **Adding categories** (eg. "Solution Evaluation")

6. The number of blogs per topic was calculated, as well as the relevance rankings per topic was averaged. Graphs were created from these results.

7. The results from Step 1 and Step 2 were compared such that the number of blogs in each type and topic were determined, as well as a Pearson correlation test was performed to measure the linear dependence between types and topics of architectural blogs.

3.4 Methodology Step 3: AK concepts

Qualitative Content Analysis (QCA) was used during Step 3 to annotate AK concepts in architectural blogs. QCA is a data analysis technique involving a qualitative step (assignment of categories to text) and a quantitative step (working through many text passages and analysis of frequencies of categories) [9]. This method resulted in information from architectural blogs to be assigned categories based on which AK concept is discussed, using annotations on Atlas.ti.

This research process is represented by a 7-step model proposed by Mayring [9]. The execution of each step during Step 3 of this project is as follows:

1. **Research question**: What AK concepts are discussed in architectural blogs?

2. **Definition of the category system (main categories and subcategories) from theory:** The state of the art and preceding studies on AK concepts was analyzed to get a theoretical foundation on the topic prior to performing the annotations. All categories used in this project, with the exception of "Quantitative Evaluation", were gathered from previous studies.

   Table 4 shows which AK concepts were present in the three studies by Soliman et al. The titles of these are `Developing an Ontology for Architecture Knowledge from Developer Communities` [12], `An Exploratory Study on Architectural Knowledge in Issue Tracking Systems` [11] and `Exploring Web Search Engines to Find Architectural Knowledge`[14].

   Soliman et al. [12] completed 3800 annotations from 105 URLS and defined 54 ontology classes (11 composite AK, 14 simple AK, 29 lexical triggers). Soliman et al. [11] performed 3,937 annotations for AK concepts, and defined 10 AK concepts. Out of the 33 AK concepts shown in Table 4, 12 were used in this project. These are indicated in this table by a green row. The AK concepts not used in this project (indicated by a white row) and the reasoning is as follows:

   - No. 1 and 3-9: These concepts are simple AK concepts and would need to be annotated per word, which was considered too detailed for this project.
   - No. 17, 18, 21 and 22: These are not applicable to blogs as they refer to an "existing system".
   - No. 23, 30 and 31: These concepts are repetitions of benefits and drawbacks.
   - No. 24, 26 and 27: These concepts did not occur often enough in blogs to be viewed as significant enough to include in this project.
   - No. 28, 32 and 33: These concepts are too simple.

Table 4: AK concepts present in three studies by Soliman et al. [11, 12, 14]. A green row indicates the AK concept was used in this project.

No.	ID	Class Name	[12]	[11]	[14]
1	TEC	Technology Solution	✓		
2	PAT	Architecture Pattern	✓		
3	QA	Quality Attribute	✓		
4	COM	Architecture Component	✓		
5	CON	Architecture Connector	✓		
6	COME	Component Element	✓		
7	COND	Connector Data	✓		
8	PROB	Software Problem	✓		
9	FT	Feature Term	✓		
10	CONF	Architecture Configuration	✓	✓	
11	CB	Component Behavior	✓	✓	
12	REQ	Requirements and Constraints	✓	✓	
13	UR	User Request	✓		
14	FEAT	Technology Features	✓		
15	ASTA	Technology Benefits and Drawbacks	✓		
16	CASE	Technology Use-Cases	✓		
17	EX	Existing System	✓		
18	DI	Design Issue	✓		
19	ADD	Recommended Design Decisions	✓		
20	DR	Decision Rules	✓		
21	EXA	Architecture of existing system		✓	
22	EXQ	Quality issues of existing system		✓	
23	ABD	Architectural solution benefits and drawbacks		✓	
24	ASSUM	Assumptions		✓	
25	TRO	Trade-offs		✓	
26	RIS	Risks		✓	
27	TAC	Architectural tactics		✓	
28		Solution Description			✓
29		Solution alternatives			✓
30		Solutions benefits			✓
31		Solutions drawbacks			✓
32		Made design decisions			✓
33		Others			✓

3. **Definition of the coding guideline (definitions, anchor examples and coding rules):** Please refer to Results (Section 4), specifically Table 19 for AK concept definitions and Table 20 for anchor examples. Note that the examples were filled in after all annotations were completed.

4. **Coding (Material run-through, preliminary codings, adding anchor examples and coding rules):** The goal was to complete roughly 1500 annotations for AK concepts. Soliman et al. [12] performed on average 36 annotations per web page. Based on that value, it was expected that Step 3 would require roughly 40 blogs to be analysed. However since forums are structured differently to blogs (eg. they have clear distinction between question and answer), to have a more accurate idea of the required sample size for Step 3, an initial sample of 13 blogs were annotated out of the set of 257 blogs from Step 2. These 13 blogs were all the blogs with relevance ranking 5 used in Step 2, and had an average of 50 annotations per blog. This initial test on a sample of blogs allowed me to gain more accurate insight into which AK concepts were most common, and hence to pay more attention to, and which were most probably not applicable to this project.

5. **Revision:** Revision of the categories and coding guideline is required to be completed after 10 - 50% of the material. The first revision was completed after the initial analysis of the sample of 13 blogs with relevance 5. Considering there were 35 blogs used in this step in total, revision was completed after 37% of the material, which aligns with the requirements. When reviewing the results throughout Step 3, constant comparison was used by checking all quotations in one concept to ensure consistency throughout the category.

   After this revision of the initial sample, revisions also took place roughly halfway through the process (not only by myself but also by my supervisor, with whom a discussion on the results was had), and once again at the end.

   While the category definitions remained unchanged, some changes in annotations and understanding of AK concept categories made during revision is as follows:

   - "Benefits and Drawbacks", "Architecture Configuration", "User Requirements" and "Component Behaviour" are largely unchanged and well defined.
   - Annotating simple AK concepts was terminated (eg. "Quality Attributes") since this was performed a word level and resulted in an excessive amount of detail.
   - Remove "Code snippets", "Quality Attribute Descriptions", "Technology Solutions", "Existing System", "Solution Description", "Design Issues" and "Tactics"
   - Add "Quantitative Evaluation" (eg. latency, performance testing)
   - Clarify the definitions of "Drawbacks" vs "Constraints" and "Benefits" vs "Features"
   - Reclassify many blogs in "Solution Description", "Recommended Design Decision" and "Requirements and Constraints" into "Benefits and Drawbacks" or "Features".

   Note that once a change was made, all quotations in that category were reviewed.

6. **Final working through the material:**

   The blogs sampled for Step 3 were collected in order of decreasing relevance.

   - **13 blogs of relevance 5**
   - **14 blogs of relevance between 4 and 5:** All blogs of relevance above 4 (including decimals) from the data set of Step 2 were annotated, with the exception of two blogs: https://www.developer.com/lang/jscript/top-7-open-source-json-binding-providers-available-today.html, which had no text on the web page, and https://geekflare.com/de/best-stock-market-api/, which was a duplicate of the first blog analysed, with the slight difference that it was in German rather than English.
   - **10 blogs of relevance 4:** There were 71 blogs with relevance of exactly 4. The 10 blogs from this data set were selected at random. Note that this resulted in no blogs of the types "University" or "Magazines and Newspapers" in the results of Step 3. This is justified as there were no "University" blogs with relevance 4 or higher, and only one "Magazines and Newspapers" blog with relevance of exactly 4.

     I continued analysing blogs of relevance 4 until the goal of over 1500 annotations was reached.

   This resulted in a total of 35 blogs analysed with 1662 annotations of AK concepts.

7. **Analysis, category frequencies and contingencies interpretation:** The definitions and IDs of the AK concepts used in Step 3 were finalised, examples of each were collected and put into a table, graphs were created for the number of annotations per concept, per type and per topic. The code co-occurrence table and code-document table from Atlas.ti were exported and analysed.

4 Results

4.1 Results Step 1: Types

The results from Step 1 contain the following:

- **Descriptions:** Table 5 contains the description and sub-categories for each of the 8 blog types concluded from Step 1.
- **Indicators:** Table 6 shows important attributes that define a blog type which were considered when categorising each blog. This helps differentiating each type. The term "Maybe" indicates that the type classification holds regardless of that attribute.
- **Examples:** Table 7 contains three examples per blog type.
- **Relevance per Type:** Figure 2 shows box plots of the relevance for each blog type.
- **Number of blogs per Type:** Table 8 shows the number of blogs and average relevance per blog type. This table does not include sub-categories. Please refer to Table A.1 in the Appendix for the table representing the "Number of Blogs and Average Relevance per blog Type including sub-categories". The results from Table 8 can be seen visually in Figure 3. It is interesting viewing this data in 2 ways: ordering by number of blogs or ordering by average relevance.

  There are 31 categories, grouped into 7 main categories of types of architectural blogs.
- **Type and Relevance Correlation:** The final results from step 1 are those of the correlation test to measure the linear dependence between types of blogs and relevance. The correlation coefficient of 0.5239089 shows a moderately strong positive correlation between `Average Relevance` and `Number of Blogs per Type`.
- **Type vs Task:** The correlation is analysed between the number of blogs in each blog type and the six tasks performed in Soliman et al. [14].

Table 5: Blog Type Descriptions

Type Name	Description	Sub-categories
Community Blog	A blog written by a community of people or software engineers. Multiple people can write and publish blog posts on the website. It can contain non-IT related content (eg. food, lifestyle, travel).	Community blog on Educational IT course provider site, Community blog on a tutorial site, Technology specific community blog
Technology Vendor	A blog run by a company that sells a product.	Analytics, API, Authentication, Books, Chatbots, Cloud, eCommerce, Finance, Healthcare, Integration
Personal blog	A blog created by a single person, developer or software engineer. The content of a personal Blog is limited to the knowledge gained from personal experience of a single person.	
IT Service Company	A blog run by a company, agency or institute providing a service such as consulting and web, software, or app development.	
Educational IT course provider	A training, learning or skills development organisation or platform providing courses and certifications that customers pay for.	
Magazines and Newspapers	A blog run by a company selling news articles not related to IT. The authors of the articles work for the company. It usually has a subscription service paid by customers on a monthly basis.	
University blog	A blog run by a university. Articles are posted by students of the university or specialists.	
Not a blog	All other websites not considered a blog. (eg. A Tutorial is a website offering step by step low level instructions on an IT related topic. This is not a blog since it does not contain AK.)	Tutorial, Scientific Journal publisher, Standards organisation, Forum, NA, LinkedIn page, Book chapter

Table 6: Important Attributes per Blog Type.

Type	Multiple Authors	Authors employed by blog company	Technology related	Vendor	Service Provider
Community Blog	Yes	No	Maybe	No	No
Personal blog	No	No	Maybe	No	No
Technology Vendor	Maybe	Yes	Yes	Yes	Maybe
IT service company	Maybe	Yes	Yes	No	Yes
Educational IT course provider	Maybe	Yes	Yes	Offers courses and/or certifications for a price	
Magazines and Newspapers	Yes	Yes	Maybe	Offers subscriptions for a price	
University Blog	Maybe	Yes	Maybe	No	No

Table 7: Examples of each Blog Type

Type	Examples
Community blog	https://dzone.com/refcardz/enterprise-integration
	https://medium.com/swlh/apache-kafka-in-a-nutshell-5782b01d9ffb
	https://www.webdesignerdepot.com/2020/09/real-time-stock-data-using-marketplaces-api/#
Technology Vendor	https://auth0.com/blog/beating-json-performance-with-protobuf/#
	https://www.redhat.com/en/blog/intro-scalability#
	https://www.informit.com/articles/article.aspx?p=1850815#
Personal blog	https://martinfowler.com/articles/microservices.html
	https://www.ben-morris.com/why-is-loose-coupling-between-services-so-important/#
	https://www.javatpoint.com/microservices#
IT Service Company	https://antaresnet.com/data_collection/#
	https://www.openlogic.com/blog/activemq-vs-rabbitmq#
	https://www.scrapehero.com/best-data-management-etl-tools/#
Educational IT course provider	https://www.educba.com/apache-flume/#
	https://www.edureka.co/blog/apache-flume-tutorial/#
	https://www.whizlabs.com/blog/real-time-data-streaming-tools/#
Magazines and Newspapers	https://analyticsindiamag.com/top-9-etl-tools-for-data-integration-in-2020/#
	https://www.opensourceforu.com/2015/12/an-introduction-to-apache-activemq/#
	http://cascadebusnews.com/advantages-disadvantages-using-mass-messaging-business/#
University	https://onlinemasters.ohio.edu/blog/big-data-analytics-tools/#
	https://hbr.org/2006/07/the-sales-learning-curve#
	https://blog.mi.hdm-stuttgart.de/index.php/2020/04/13
Not a blog	https://www.tutorialspoint.com/apache_camel/apache_camel_overview.htm#
	https://freecontent.manning.com/strategies-for-decomposing-an-application-into-services/#
	https://journals.sagepub.com/doi/10.1155/2016/2415016#

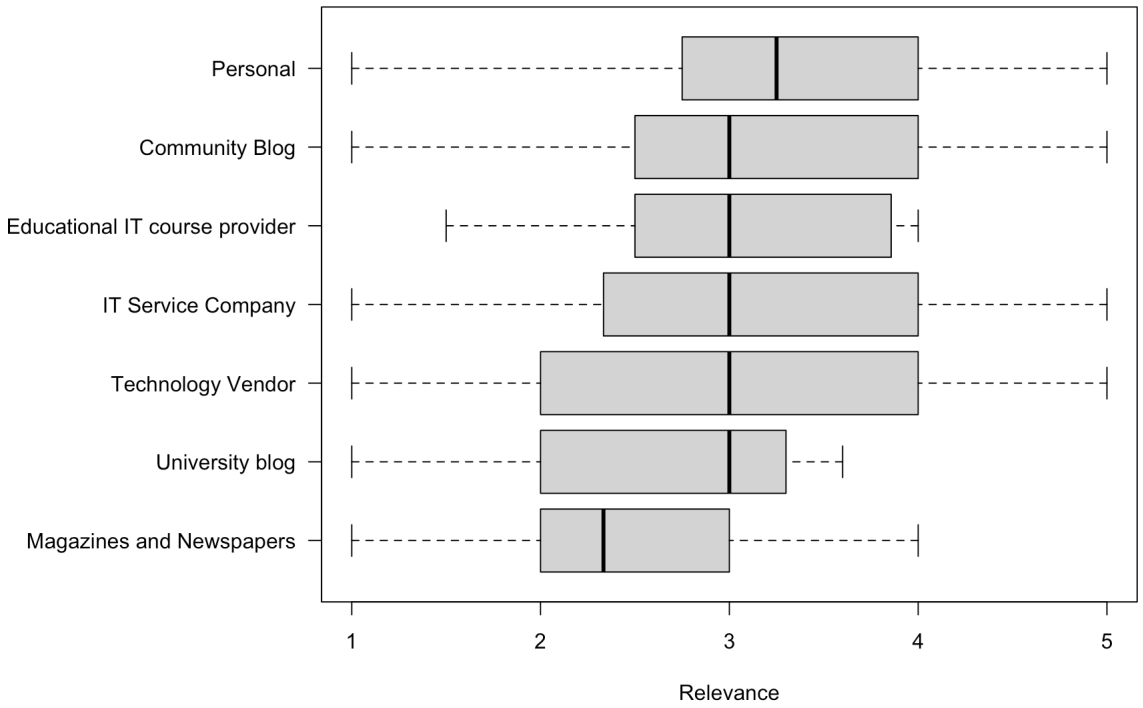
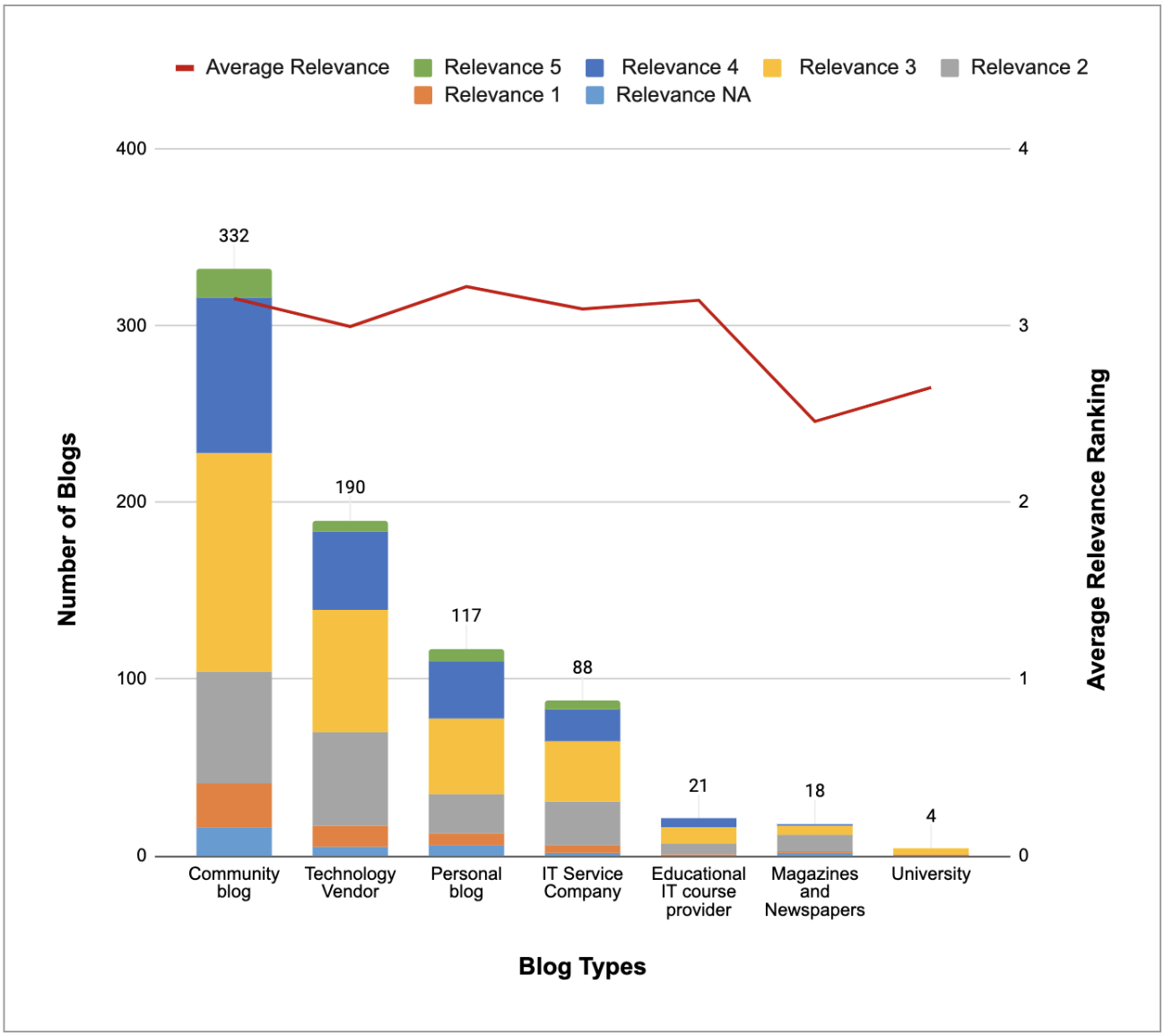


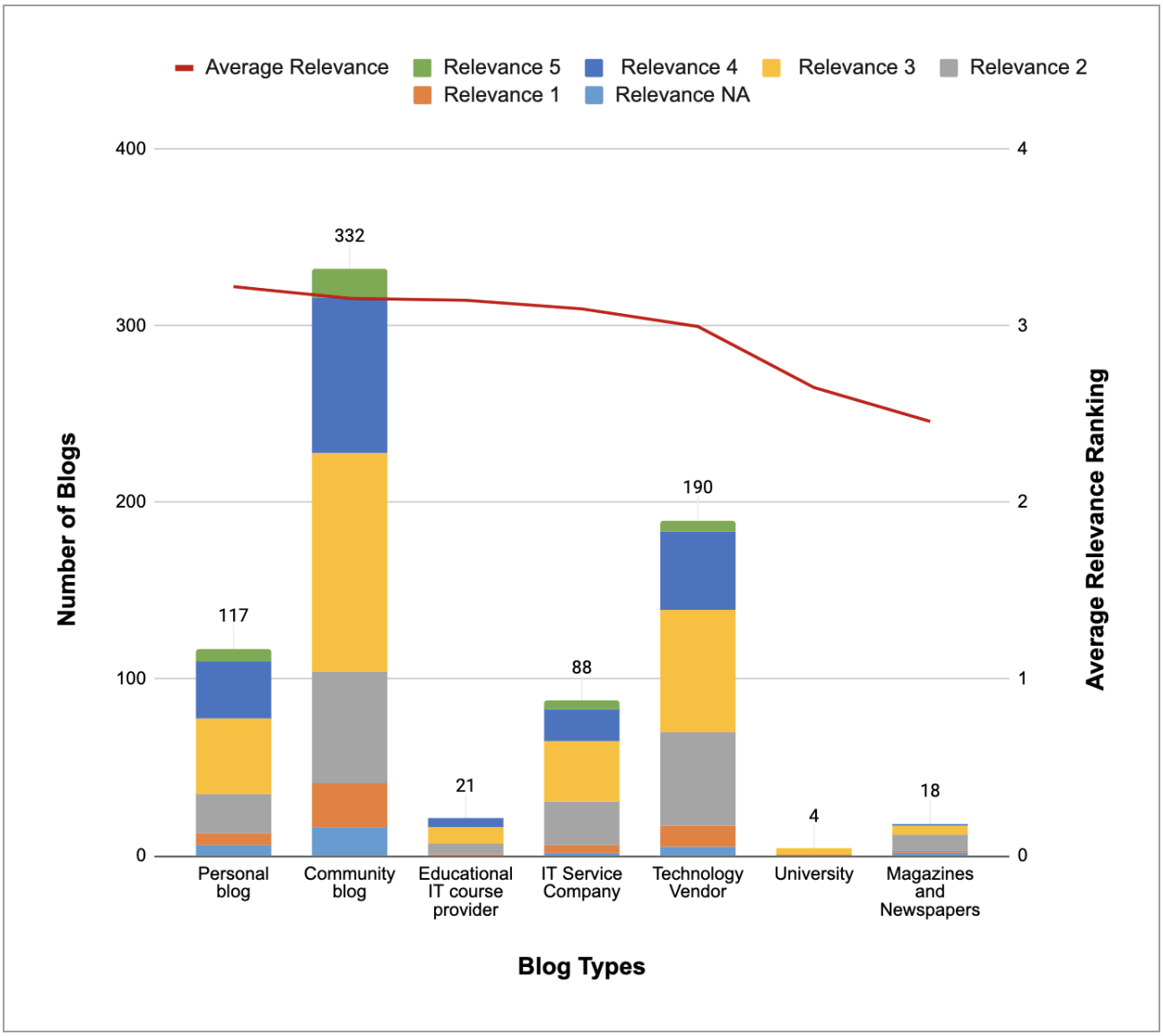
Figure 2: Box plots representing the Relevance of each Blog Type.

Table 8: Number of Blogs in each Relevance Ranking and Average Relevance per Blog Type excluding sub-categories from Step 1.

Type	**Average Relevance**	**Total**	**Relevance**					
			NA	**1**	**2**	**3**	**4**	**5**
Personal blog	3.221088382	117	6	7	22	43	32	7
Community blog	3.154718426	332	16	25	63	124	88	16
Educational IT course provider	3.143613001	21	0	1	6	9	5	0
IT Service Company	3.094832251	88	2	4	25	34	18	5
Technology Vendor	2.995225313	190	5	12	53	69	45	6
University	2.65	4	0	1	0	3	0	0
Magazines and Newspapers	2.458333333	18	2	1	9	5	1	0



(a) Ordered by decreasing Number of Blogs.



(b) Ordered by by decreasing Average Relevance Ranking

Figure 3: Graph of Relevance Ranking and Number of Blogs per Blog Type.

4.1.1 Type vs Task

Table 9 shows the number of blogs in each blog type and in each of the six tasks retrieved by participants of Soliman et al. [14]. Table 10 merges the tasks for the same Attribute Driven Design step, these are:

- Messaging Evaluation and Big-Data-Stream-Evaluation
- Conceptual design and Physical design
- Middleware search and JSON search

Table 11 and 12 shows these same values, however only for blogs with a Relevance Ranking of 3 and above.

Chi-square tests for independence were performed on the results of Table 12. A python program was written to calculate these values. First a contingency table was created, and then the `chi2_contingency` function was executed for each cell of the table. This shows the the co-occurrences between ADD tasks and blog types. The cells with a value of higher than 10 are considered significant.

Table 9: Number of blogs per Type and Task performed in Soliman et al. [14]

Type	**Tasks**						
	Big-Data -Stream -Evaluation	**Messaging -Evaluation**	**Conceptual -Design**	**Physical -Design**	**Middleware -Search**	**JSON -Search**	**NA**
Community blog	78	98	79	24	34	38	8
Technology Vendor	47	39	31	35	31	14	6
Personal	11	26	21	17	9	33	6
IT Service Company	26	18	21	6	15	6	1
Educational IT course provider	8	7	6	0	3	6	0
Magazines and Newspapers	8	3	3	0	3	1	0
University blog	3	1	0	0	0	0	0
Total	**181**	**192**	**161**	**82**	**95**	**98**	**21**

Table 10: Table 9 grouped into Task Number

Type	**Tasks**			
	1	**2**	**3**	**NA**
Community blog	176	103	72	8
Technology Vendor	86	66	45	6
Personal	37	38	42	6
IT Service Company	44	27	21	1
Educational IT course provider	15	6	9	0
Magazines and Newspapers	11	3	4	0
University blog	4	0	0	0
Total	**373**	**243**	**193**	**21**

Table 11: Table 9 with only blogs with a relevance of 3 and above

Type	Tasks						
	Big-Data -Stream -Evaluation	**Messaging -Evaluation**	**Conceptual -Design**	**Physical -Design**	**Middleware -Search**	**JSON -Search**	**NA**
Community blog	43	72	50	14	23	25	1
Educational IT course provider	1	2	3	0	3	5	0
IT Service Company	18	16	10	4	4	5	0
Magazines and Newspapers	4	1	1	0	0	0	0
Personal	4	21	11	14	7	23	2
Technology Vendor	27	27	18	26	12	10	0
University blog	3	0	0	0	0	0	0
Total	**100**	**139**	**93**	**58**	**49**	**68**	**3**

Table 12: Table 11 grouped into task Numbers

Type	Task			
	1	**2**	**3**	**NA**
Community blog	115	64	48	1
Educational IT course provider	3	3	8	0
IT Service Company	34	14	9	0
Magazines and Newspapers	5	1	0	0
Personal	25	25	30	2
Technology Vendor	54	44	22	0
University blog	3	0	0	0
Total	**239**	**151**	**117**	**3**

Table 13: Chi-squared test matrix for blog types and tasks

Type	Task					
	Big-Data-Stream-Evaluation	**Messaging-Evaluation**	**Conceptual-Design**	**Physical-Design**	**Middleware-Search**	**JSON-Search**
Community blog	0	3	3	10	0	1
Educational IT course provider	0	0	0	0	1	4
IT Service Company	4	0	0	0	0	0
Magazines and Newspapers	5	0	0	0	0	0
Personal	11	0	0	2	0	17
Technology Vendor	0	1	0	14	0	2
University blog	7	0	0	0	0	0

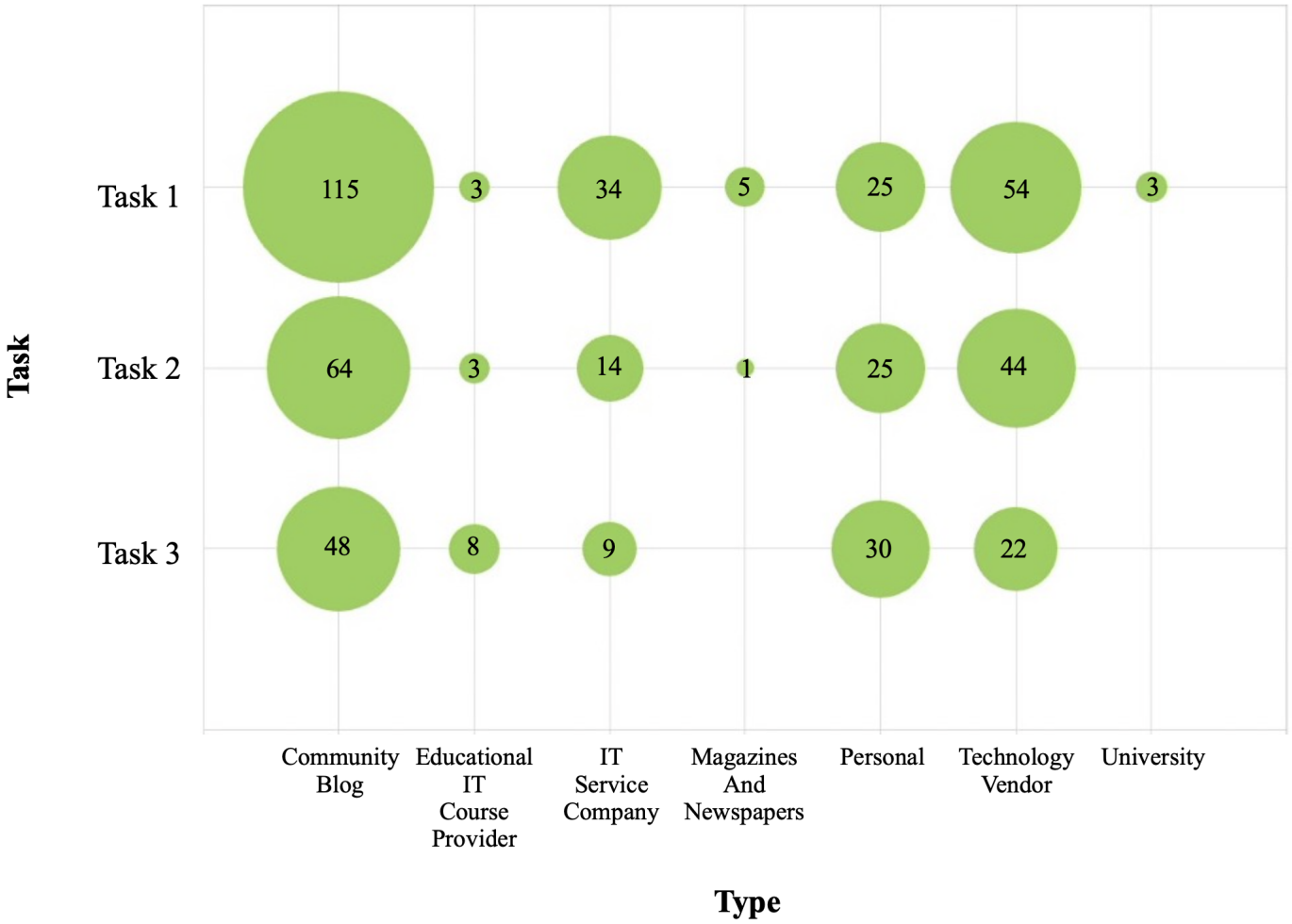


Figure 4: Bubble graph of Table 12 showing the number of blogs per blog Type and Task performed in Soliman et al. [14]

4.2 Results Step 2: Topics

The results from Step 2 contain the following:

- **Descriptions and indicators:** Table 14 contains the 5 blog topics found in Step 2, along with their definitions and common indicators. The term "solution" in this table refers to any of the following items: technology, concept, method, application, attribute, tools, patterns, features, characteristics.

  Figure 5 explains how to distinguish between a "Use Case" and a "How To" blog. When 4 layers of abstraction are considered in an architectural system, an architectural blog describing the basic functionalities of how to implement a technology, the blog is a "How To" blog. If the blog describes any form of requirements or conceptual solutions, the blog is a "Use Case" blog.

- **Examples:** Table 16 contains three examples per blog topic.

- **Relevance per Topic:** Figure 7 shows box plots of the relevance for each blog topic.

- **Number of blogs per Topic:** Table 15 shows the the number of blogs and average relevance for each blog topic found in Step 2. This data is visualised in Figure 6. Similarly to the results of Step 1, I have represented these results in 2 sub-graphs, ordering the x axis by the 2 different variables.

- **Type vs Topic:** The correlation between blog types from Step 1 and blog topics from Step 2 is calculated.

Table 14: Blog Topic Definitions

Topic	**Definition**	**Common Indicators**
List of alternative solutions	Lists solutions often with a short summary or description of each element in the list. Sub-categories are "Best Practices", "tools" and "patterns".	Top [number] [solution]
Comparison	Compares two or more solutions. Often states pros and cons, provides a recommendation for when to use or not use a solution, or compares solutions based on performance, speed or latency.	[solution] vs [solution]
Solution Evaluation	Explain, describe, discuss or define features (eg. benefits, design principles) to summarise or evaluate a solution.	What is [solution], How does [solution] work, What does [solution] do, Benefits of [solution]
Use-case	Proposes a conceptual solution, possibly using a specific technology, given system requirement(s).	How to build, code or develop [an application] with [a technology].
How to	Explains the implementation of a specific technology, usually with code snippets.	How to use [technology].

Table 15: Number of blogs and Average Relevance per Blog Topic

Topic	**Count**	**Average relevance**
How to	18	3.518518519
Use Case	43	3.133133146
Comparison	69	3.428888834
Solution Evaluation	78	3.148527808
List of alternative solutions	18	3.310606061
List of alternative solutions: Best practices	7	3
List of alternative solutions: Patterns	8	3.458333333
List of alternative solutions: Tools	16	3.069791667
List: Total	49	3.211719233

Table 16: Blog Topic Examples

Topic	Examples
Solution Evaluation	https://www.infoq.com/articles/modular-java-what-is-it/#
	https://www.janbasktraining.com/blog/what-is-flume/#
	https://nordicapis.com/all-you-need-to-know-about-rest-api-design/#
Comparison	https://linuxhint.com/rabbitmq-vs-apache-kafka/#
	https://tanzu.vmware.com/content/blog/understanding-when-to-use-rabbitmq-or-apache-kafka#
	https://dattell.com/kafka-vs-rabbitmq-how-to-choose-an-open-source-message-broker
List	https://blog.panoply.io/17-great-etl-tools-and-the-case-for-saying-no-to-etl#
	https://blog.todotnet.com/2017/07/design-patterns-for-microservices/#
	https://www.datadriveninvestor.com/2019/02/25/6-alternatives-to-the-yahoo-finance-api/#
Use Case	https://dzone.com/articles/stans-robot-shop-a-sample-microservice-application
	https://eclipsesource.com/blogs/2013/04/18/minimal-json-parser-for-java/#
	https://medium.com/build-a-chat-application-using-spring-boot-websocket-rabbitmq
How to	https://crunchify.com/how-to-read-json-object-from-file-in-java/#
	http://blog.florian-hopf.de/2019/07/apache-camel.html#
	https://hackernoon.com/connecting-rabbitmq-with-node-js-05953yh3#

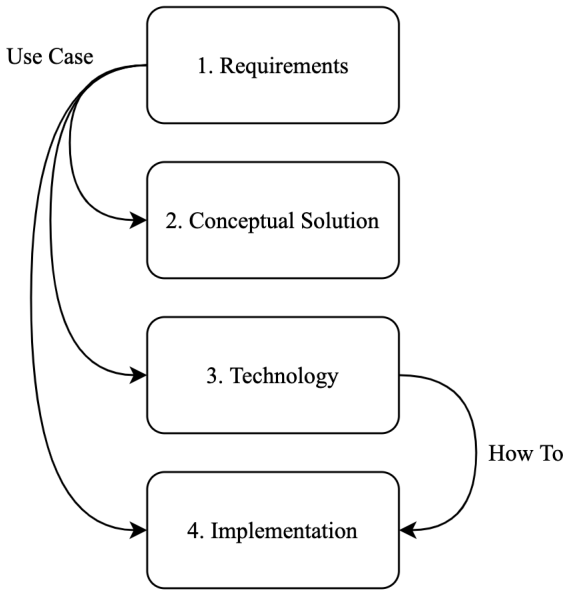
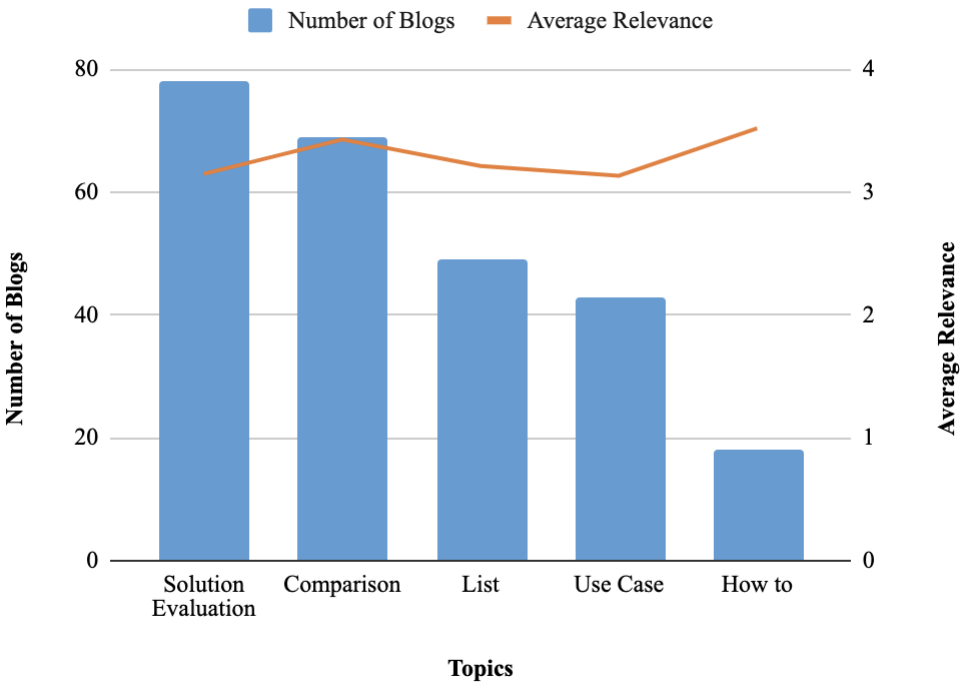
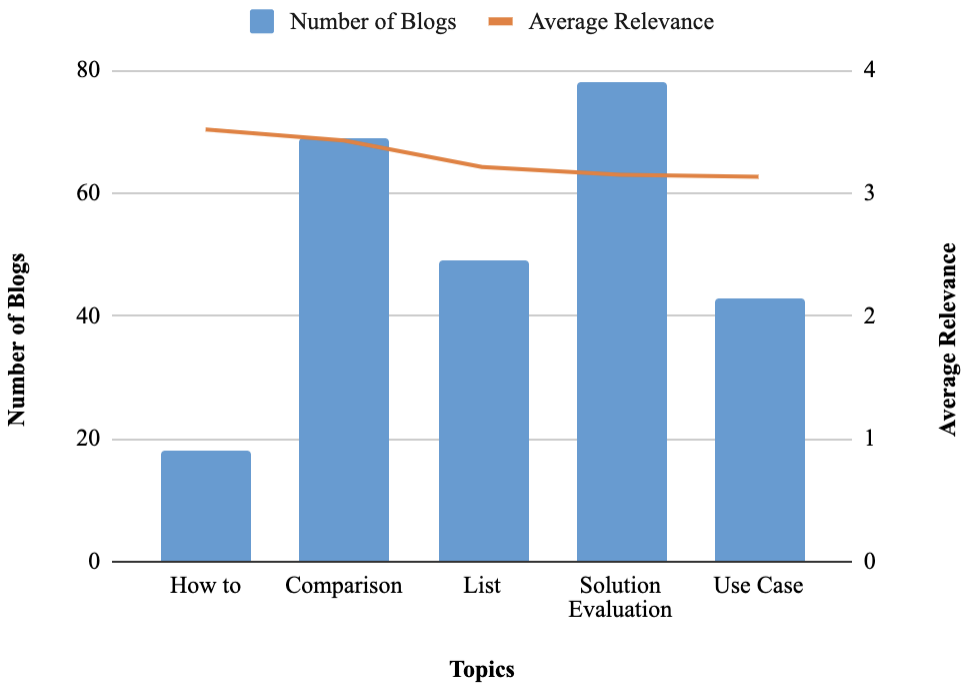


Figure 5: Figure showing the distinguishing factor between Blog Topics “Use Case” and “How To”.



(a) Ordered by decreasing Number of Blogs



(b) Ordered by decreasing Average Relevance Ranking

Figure 6: Graph of Relevance Ranking and Number of Blogs per Blog Topic.

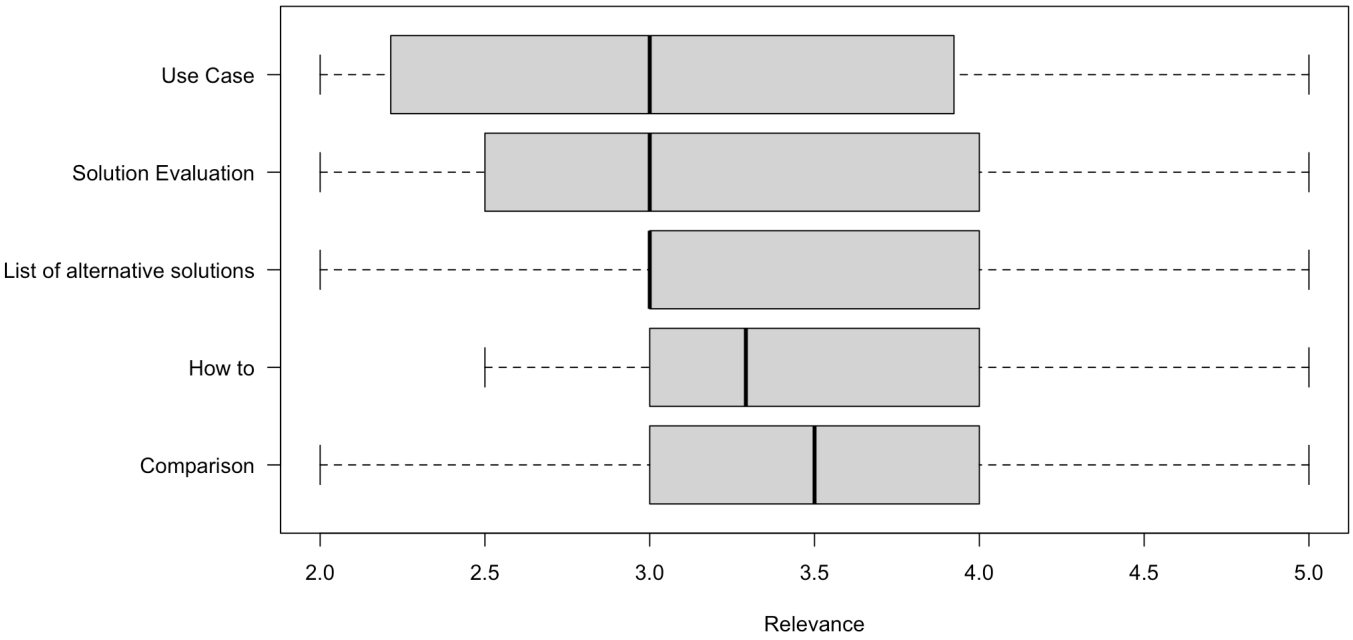


Figure 7: Box plots representing the Relevance of each Blog Topic.

4.2.1 Type vs Topic

Table 17 shows the number of blogs in each type and topic assigned in Steps 1 and 2, excluding sub-categories.

A Chi-square test for correlation between blog type and topic was computed in R using *chisq.test()*. The warning "Chi-squared approximation may be incorrect" appeared, meaning that the smallest expected frequencies were lower than 5, so I used the Fisher's exact test as it does not require the assumption of a minimum of 5 expected counts in the contingency table. This test resulted in a Chi-squared value of 26.847, and p-value of 0.3123. Therefore is a high Chi-squared value and a p-value of greater than 0.05 significance level. Therefore the correlation between blog type and topic cannot be considered significant.

Table 17: Number of Blogs per Blog Type and Blog Topic

	Topics				
Type	**Comparison**	**How to**	**Use Case**	**Solution Evaluation**	**List**
Educational IT course provider	1	0	1	4	1
IT Service Company	8	1	9	7	5
Magazines and Newspapers	0	0	1	2	3
Personal	12	5	9	9	4
University blog	0	0	0	0	1
Community blog	29	6	17	34	24
Technology Vendor	19	6	5	22	11

4.3 Results Step 3: AK concepts

The results from Step 3 contain the following:

- **Data set:**
  - **AK concepts:** The initial set of AK concepts was gathered from Soliman et al. [11, 12, 14], however as explained in the Methodology in Section 3.1, not all concepts mentioned in those studies were relevant to blogs and hence not used in this project. Note that "QEV" was created and was not present in the studies from Soliman et al. [11, 12, 14].
  - **Documents:** 35 blogs were uploaded to Atlas.ti as documents and annotated. Each document's URL can be found in Table A.1 in the Appendix, while each document's `Relevance Ranking, Type, Topic, Number of Quotations` and `Number of Codes` can be found in Table 18. The Number of Codes per document can be visualised in the box plot in Figure 8.
- **Qualitative Results:**
  - **Definitions:** Table 19 lists the 13 AK concepts annotated during Step 3, their ID's for easier identification and their definitions. This will help other researchers to annotate AK concepts in blogs in future research.
  - **Examples:** Table 20 shows three examples per AK concept, including their respective `Quotation Number` and `File Name`. Note that the `Quotation Number` "$x : y$ ¶z" refers to document number x, quotation number y, and paragraph number z.
- **Quantitative Results:**
  - **Number of blogs per AK concept:** Table 21 and Figure 9 show the number of annotations made in Atlas.ti of each AK concept in Step 3. Further detail is added to these results by specifying the distribution of each type and topic within each AK concept. This distribution is shown in Tables 26 and 27, and Figures 10 and 11.
  - **Number of AK concepts per Type:** Refer to Section 4.3.1 which contains Table 24 and Figure 12.
  - **Number of AK concepts per Topic:** Refer to Section 4.3.2 which contains Table 25 and Figure 13.
  - **Code Co-occurrence table:** This table shows how often a combination of two codes was linked to the same quotation.
  - **Code-Document table:** This table shows the amount of each AK concept annotated in each document in Atlas.ti in Step 3.

Table 18: Relevance Ranking, Type, Topic, Number of Quotations and Number of Codes in each Document annotated in Atlas.ti in Step 3.

Doc No.	Relevance	Type	Topic	Quotations	Codes
1	5	Technology Vendor	List of alternative solutions	72	73
2	5	Personal	Comparison	59	64
3	5	IT Service Company	Comparison	53	54
4	5	Community blog	Use Case	15	15
5	5	Community blog	Comparison	32	32
6	5	IT Service Company	Comparison	45	45
7	5	Community blog	Use Case	40	40
8	5	Community blog	Use Case	20	20
9	5	Technology Vendor	Solution Evaluation	106	106
10	5	Personal	Comparison	107	109
11	5	Personal	Comparison	19	19
12	5	Community blog	Solution Evaluation	16	16
13	5	Community blog	How to	54	54
14	4.666666667	IT Service Company	Comparison	47	49
15	4.5	Technology Vendor	Comparison	34	35
16	4.5	Community blog	How to	15	15
17	4.5	Technology Vendor	Use Case	45	46
18	4.333333333	Community blog	Solution Evaluation	81	82
19	4.333333333	IT Service Company	Use Case	13	13
20	4.264705882	Technology Vendor	Comparison	114	116
21	4.5	Community blog	Solution Evaluation	43	44
22	4.25	Community blog	Solution Evaluation	77	82
23	4.066666667	Community blog	Comparison	16	16
24	4.111111111	Personal	Comparison	15	15
25	4.052631579	Technology Vendor	Comparison	42	44
26	4	Technology Vendor	List of alternative solutions	72	72
27	4	Personal	List of alternative solutions	41	42
28	4	IT Service Company	Solution Evaluation	21	21
29	4	IT Service Company	How to	40	40
30	4	Educational IT course provider	Solution Evaluation	26	27
31	4	Educational IT course provider	Solution Evaluation	21	22
32	4	Technology Vendor	Solution Evaluation	95	96
33	4	Technology Vendor	Solution Evaluation	47	49
34	4	Community blog	Solution Evaluation	37	37
35	4	Community blog	List of alternative solutions	52	52
		TOTAL		**1632**	**1662**

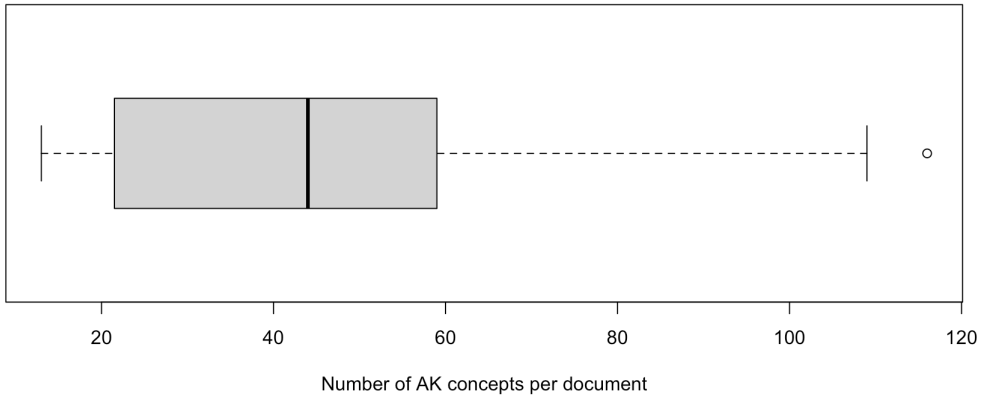


Figure 8: Box plot of number of AK concept codes per Document in Step 3.

Table 19: Definitions of 13 AK concepts annotated during Step 3

ID	AK Concept	Definition
CONF	**Architecture Configuration**	Describes the dependencies of components. It represents part of an architectural model, which consists of one or more component names associated with an architecture connector verb or name. [11, 12]
PAT	**Architecture Pattern**	A general, reusable solution to a commonly occurring problem in software architecture within a given context.
CB	**Component Behavior**	Describes the behavior of an architecture component. It gives an overview about the type of implemented logic and complexity. Sometimes internal operations are mentioned during the description. [12]
DR	**Decision Rules**	Conditional recommendation for architectural solutions. The rule condition might involve other ontology classes such as Recommended Design Decisions and Requirements and Constraints. [12]
QEV	**Quantitive evaluation**	An evaluation method that yields numerical indices gathered primarily from formal objective methods of data collection, systematic and controlled observation, and a prescribed research design.
ADD	**Recommended Design Decisions**	Recommendations from users based on their experience or opinion for certain architectural solutions. [12]
REQ	**Requirements and Constraints**	Requirements include: 1) Quality Attribute Requirements, such as performance, maintainability, security, 2) User Functional Requirements, such as use cases and user stories, 3) Technology Features Requirements. Constraints include: 1) Contextual Constraints, such as external systems or constraints from managers, 2) Technical Skills Sonstraint, 3) Development Time Constraint, 4) Solution Constraint. [11, 12]
ALT	**Solution alternatives**	Multiple (alternative) architectural options for a certain design issue. The architectural options could be listed in the text or as a comparison of different options. [14]
ASTA	**Technology Benefits and Drawbacks**	Technology Benefits and Drawbacks are distinguished through the extensive usage of adjectives and adverbs in combination with Technology Features and Quality Attributes. The adjectives or adverbs are used to express the advantages or disadvantages of certain technology solutions or features. [12]
FEAT	**Technology Features**	There are two types of Technology Features: 1) Development Features are expressed through certain programming activities (e.g. debugging) or programming features and tools (e.g. code generation), and 2) Behavioral Features are expressed through technology specific component and class names, as well as their implemented architectural patterns or their relationship with other technologies. [12]
CASE	**Technology Use-Cases**	Either success or failure stories for the usage of technology solutions at certain contexts. The stories could be coming from personal experiences of users, or well-known examples for existing systems. The context associated with stories could include domain description, architecture configurations, infrastructure, and constraints. [12]
TRO	**Trade-offs**	Describe balanced analysis of what is an appropriate option after prioritizing and weighing different design options. [11]
UR	**User Request**	Exist in the form of questions or needs.[12]

Table 20: AK Concept Examples

ID	Sentence	URL
CONF	Messages are pushed from RabbitMQ to the consumer.	https://www.cloudamqp.com/blog/when-to-use-rabbitmq-or-apache-kafka.html
	When it receives a request, the API gateway consults a routing map that specifies which service to route the request to.	https://dzone.com/articles/microservice-architecture -and-design-patterns-for
	Data comes into the system via a source and leaves via a sink.	https://www.upsolver.com/blog/popular-stream-processing-frameworks-compared
PAT	There are 2 main patterns of messaging: 1. queuing 2. publish-subscribe	https://freshcodeit.com/blog-introduction-to-message -brokers-part-1-apache-kafka-vs-rabbitmq
	The API Gateway Pattern is used to abstract the communication between client applications and internal microservices.	https://www.ibm.com/cloud/blog/rapidly-developing-applications-part-6-exposing -and-versioning-apis
	ActiveMQ message patterns include PUB-SUB and message queue. RabbitMQ general message patterns include, Message Queue, PUB-SUB and RPC and Routing.	https://www.openlogic.com/blog/activemq-vs-rabbitmq
CB	the Spring WebSocket application acts as the STOMP broker to clients.	https://medium.com/build-a-chat-application-using-spring-boot-websocket-rabbitmq
	The broker is responsible to send, receive, and store messages into the disk.	https://dzone.com/articles/develop-a-java-app-with-kafka
	This component is responsible for balancing the services on nodes and identifying failures.	https://www.edureka.co/blog/microservice-architecture/#
DR	Alternatives like Kafka can be used if more real-time data streaming is needed.	https://dzone.com/articles/apache-flume-and-data-pipelines
	If you want an open-source Big Data ETL, the CloverDX and Talend can be a wise choices.	https://hevodata.com/learn/best-big-data-etl-tools/
	If you want a simple/traditional pub-sub message broker, the obvious choice is RabbitMQ,	https://www.cloudamqp.com/blog/when-to-use-rabbitmq-or-apache-kafka.html
QEV	With a message rate of about 20k+ msgs/sec which is much less than Kafka, it's sufficient enough for most use cases.	http://ravindranaik.com/which-messaging-queue-is-best/#
	On average, each message had an overhead of 9 bytes in Kafka, versus 144 bytes in ActiveMQ.	https://www.infoq.com/articles/apache-kafka/#
	ZeroMQ is capable of sending over 5,000,000 messages per second but is only able to receive about 600,000/second. In contrast, nanomsg sends shy of 3,000,000/second but can receive almost 2,000,000.	https://bravenewgeek.com/tag/activemq/#
ADD	I think a ZooKeeper-free implementation is the best choice.	https://www.kai-waehner.de/apache-kafka-versus-pulsar-event-streaming-comparison-features-myths-explored/
	Apache Kafka is a perfect fit	https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/
	StreamSet is not recommended.	https://hevodata.com/learn/best-big-data-etl-tools/
REQ	All other requirements such as security, throttling, caching, monetization, and monitoring have to be done at the gateway layer.	https://wso2.com/whitepapers/microservices-in-practice-key-architectural-concepts-of-an-msa
	your requirements require a three-tier architecture	https://www.kai-waehner.de/apache-kafka-versus-pulsar-event-streaming-comparison-features-myths-explored/
	the following five constraints must be present for any application to be considered RESTful: Client-server...Statelessness...Caching...Layered system...Uniform interface...	https://blog.feathersjs.com/design-patterns-for-modern-web-apis

Table 20 continued

ID	Sentence	URL
ALT	the most popular and commonly used technologies are: 1. Python ... 2. Pandas ... 3. Twilio ... 4. TensorFlow ... 5. SpaCy ... 6. Telegram, Viber, or Hangouts APIs ...	https://sloboda-studio.com/blog/how-to-use-nlp-for-building-a-chatbot/
	there are plenty of other RabbitMQ alternatives that are viable options. For example, Apache Kafka and ActiveMQ are similar tools to RabbitMQ, and even Redis	https://www.fasthosts.co.uk/blog/rabbitmq-and-message-brokers/
	Here's the lineup for best overall stock market APIs: 1. Alpha Vantage 2. Xignite 3.Polygon.io 4. Intrinio 5.IEX Cloud 6.Tradier (and other brokerages)	https://towardsdatascience.com/best-free-and-paid-stock-market-apis-for-2020
ASTA	this can save a lot of money (e.g., manufacturing), increase revenue (e.g., vending machines) or increase customer experience (e.g., telco network failure prediction)	https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/
	Apache Flume is an efficient, distributed, reliable, and fault-tolerant data-ingestion tool.	https://dzone.com/articles/apache-flume-and-data-pipelines
	various strategies are very time-consuming, resource-intensive, and inefficient.	https://datavirtuality.com/blog-etl-tools-and-processes/#
FEAT	features like clustering, caching, logging, and message storage.	https://www.openlogic.com/blog/activemq-vs-rabbitmq
	It is built on Eclipse graphic environment. Talend supports cloud and on-premise databases. It offers a connector to other software as Saas. It offers a smooth workflow and can be adapted easily. You can deploy it on the cloud.	https://hevodata.com/learn/best-big-data-etl-tools/
	Spark's in-memory data processing engine conducts analytics, ETL, machine learning and graph processing on data in motion or at rest. It offers high-level APIs	https://www.upsolver.com/blog/popular-stream-processing-frameworks-compared
CASE	In this post, we'll introduce you to the basics of Apache Kafka and move on to building a secure, scalable messaging app with Java and Kafka.	https://dzone.com/articles/develop-a-java-app-with-kafka
	In our retail use case, you can find that we have split the capabilities of its monolith into four different microservices ... They are addressing a limited, but focused business scope, so that each service is fully decoupled from each other and ensures agility in development and deployment.	https://wso2.com/whitepapers/microservices-in-practice-key-architectural-concepts-of-an-msa/
	Many well-known and successful projects already rely on ZooKeeper. Just a few of them include HBase, Hadoop 2.0, Solr Cloud, Neo4J, Apache Blur (incubating), and Accumulo.	https://www.infoq.com/articles/apache-kafka/#
TRO	This may simplify code, but also means developers need to plan their architecture carefully to avoid inefficient processing.	https://www.upsolver.com/blog/popular-stream-processing-frameworks-compared
	Non-blocking APIs scale better, but are more complicated to design and use. Blocking APIs allow for retrying when the resource becomes available.	https://www.ibm.com/cloud/blog/rapidly-developing-applications-part-6-exposing-and-versioning-apis
	There is no cross-communication between nodes. It makes this trade-off in the name of simplicity.	https://bravenewgeek.com/tag/activemq/#
UR	How should I balance cohesion and coupling when designing software systems?	https://devopedia.org/cohesion-vs-coupling#
	Can and should Apache Kafka replace a database? How long can and should I store data in Kafka?	https://www.kai-waehner.de/apache-kafka-versus-pulsar-event-streaming-comparison-features-myths-explored/
	Is there any reason to use RabbitMQ over Kafka?	https://www.cloudamqp.com/blog/when-to-use-rabbitmq-or-apache-kafka.html

Table 21: Number of annotations per AK Concept

AK Concept	Frequency
Technology Benefits and Drawbacks	377
Technology Features	291
Requirements and Constraints	144
Component Behaviour	139
Recommended Design Decisions	116
User Request	91
Decision Rules	89
Architecture Configuration	88
Architecture Pattern	85
Technology Use Cases	83
Solution Alternatives	82
Trade-Offs	48
Quantitative Evaluation	29
TOTAL	**1662**

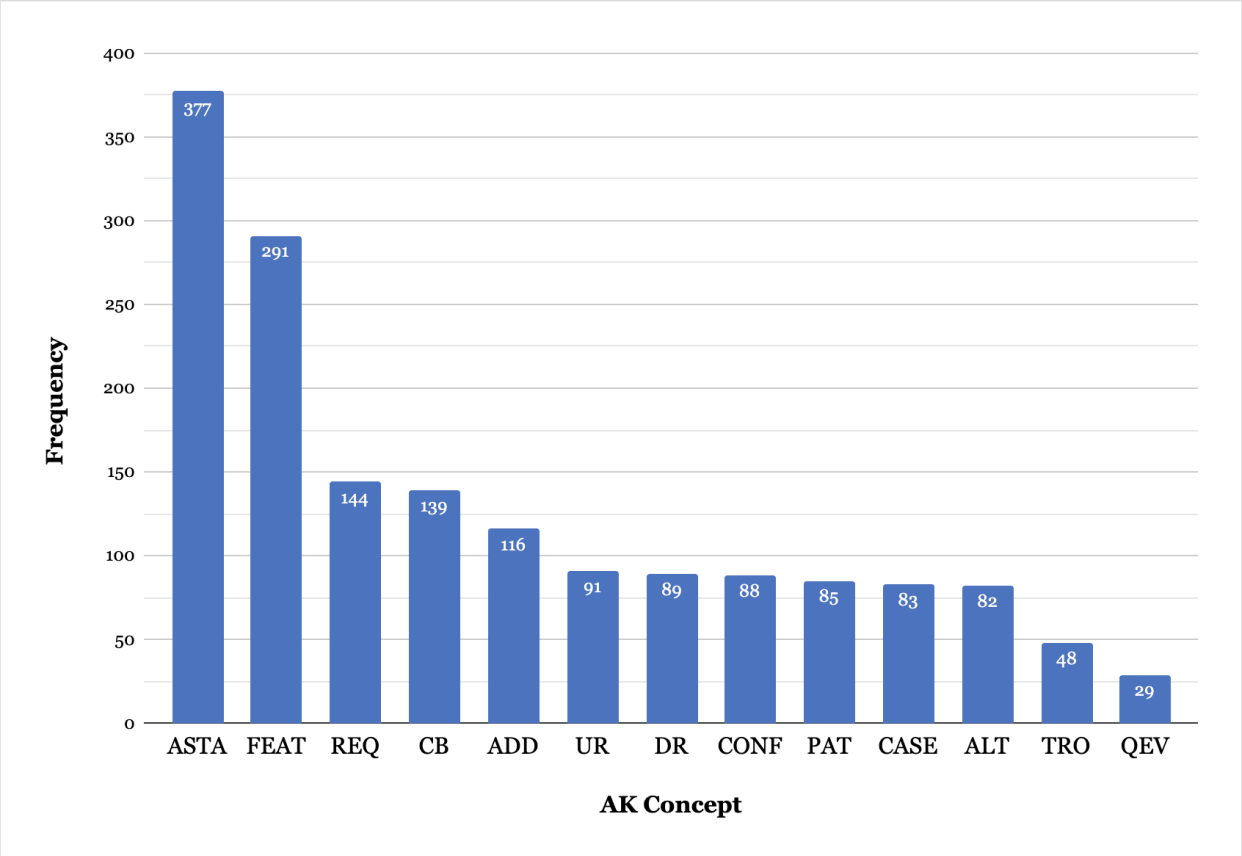


Figure 9: Number of annotations per AK Concept

The following two pages show the results of Table 21 and Figure 9 including the distribution of each blog Type and Topic.

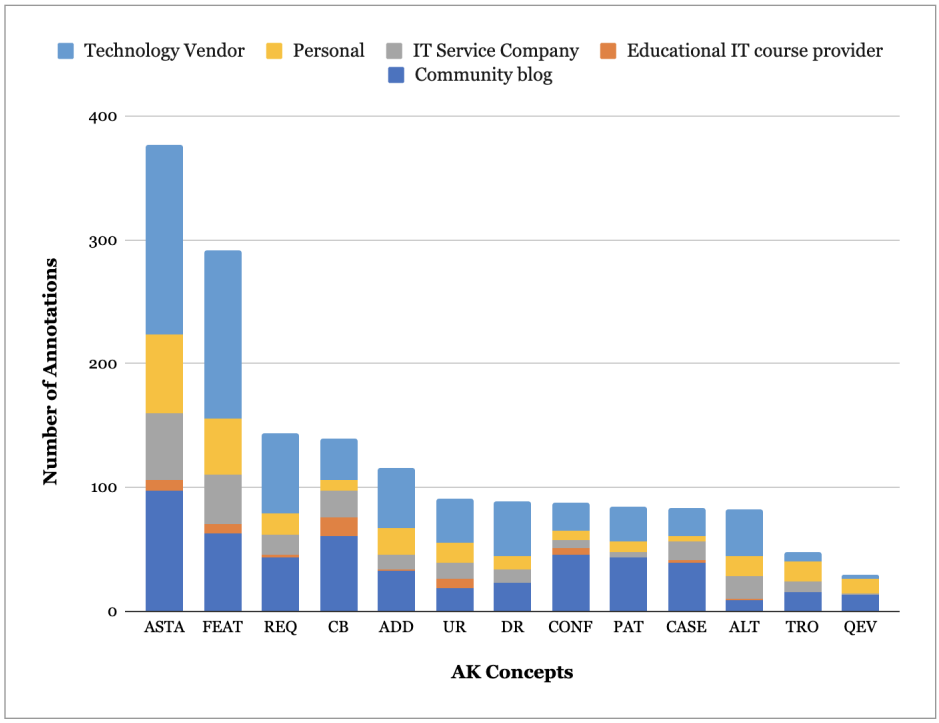


Figure 10: Distribution of blog Types in each AK concept

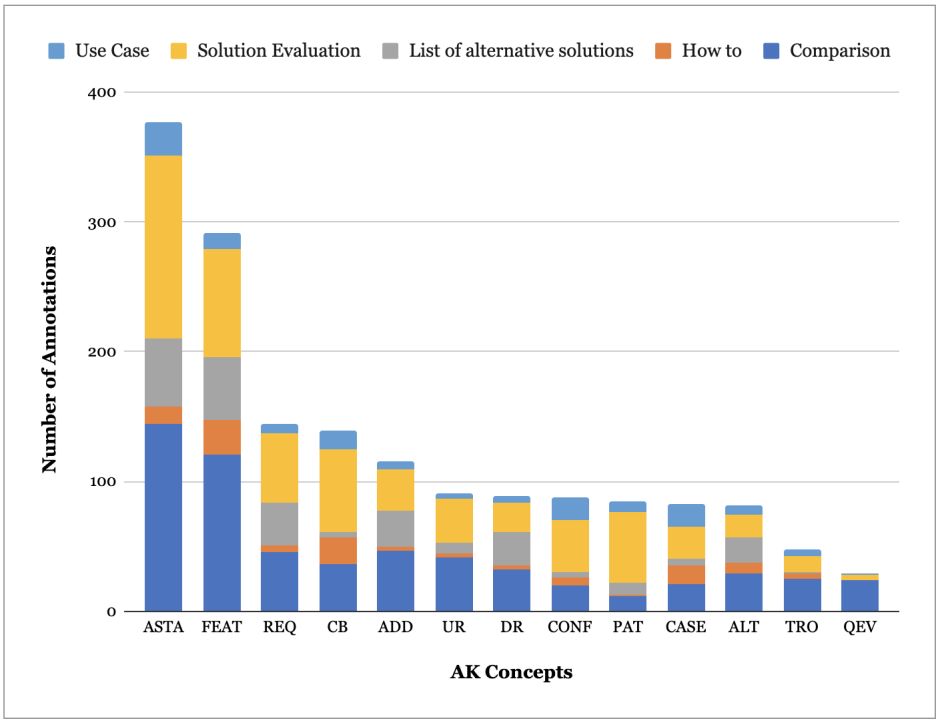


Figure 11: Distribution of blog Topics in each AK concept

Table 22: Code Co-Occurrence Table

	CONF	PAT	CB	DR	QEV	ADD	REQ	ALT	ASTA	FEAT	CASE	TRO	UR
UR	1	2	0	0	0	0	1	0	3	1	2	0	0
TRO	1	4	1	1	2	2	3	1	17	10	5	0	0
CASE	14	6	19	4	6	2	14	8	21	10	0	5	2
FEAT	7	16	19	5	2	7	7	27	107	0	10	10	1
ASTA	3	26	5	16	5	12	18	8	0	107	21	17	3
ALT	2	5	0	1	1	0	1	0	8	27	8	1	0
REQ	1	12	2	79	0	2	0	1	18	7	14	3	1
ADD	1	9	0	68	0	0	2	0	12	7	2	2	0
QEV	0	0	1	0	0	0	0	1	5	2	6	2	0
DR	1	6	0	0	0	68	79	1	16	5	4	1	0
CB	21	18	0	0	1	0	2	0	5	19	19	1	0
PAT	14	0	18	6	0	9	12	5	26	16	6	4	2
CONF	0	14	21	1	0	1	1	2	3	7	14	1	1

Table 23: Chi-square test matrix for code co-occurrence

	CONF	PAT	CB	DR	QEV	ADD	REQ	ALT	ASTA	FEAT	CASE	TRO	UR
UR	0	0	0	0	1	0	0	0	0	0	0	0	2
TRO	0	0	0	4	1	0	0	0	10	0	0	0	0
CASE	14	1	22	8	13	4	0	2	0	3	9	0	0
FEAT	0	0	2	25	0	5	12	47	179	46	3	0	0
ASTA	6	1	7	9	1	2	1	0	59	179	0	10	0
ALT	0	0	2	5	0	3	3	1	0	47	2	0	0
REQ	4	0	5	255	0	7	16	3	1	12	0	0	0
ADD	2	0	6	271	0	7	7	3	2	5	4	0	0
QEV	0	0	0	1	0	0	0	0	1	0	13	1	1
DR	7	6	12	29	1	271	255	5	9	25	8	4	0
CB	74	16	4	12	0	6	5	2	7	2	22	0	0
PAT	12	10	16	6	0	0	0	0	1	0	1	0	0
CONF	2	12	74	7	0	2	4	0	6	0	14	0	0

4.3.1 Type vs AK concepts

Table 24 and Figure 12 show the number of AK concepts per blog type. "Magazines and Newspapers" and "University" have the value zero since there were no blogs of those types in the 35 blogs annotated during Step 3. Refer to Section 3.4 for an explanation of the sampling for this Step.

Table 24: Number of AK concept annotations in each Blog Type

Type	Count	Annotations	Annotations per Blog
Technology Vendor	9	637	70.77777778
IT Service Company	6	222	49.8
Personal	5	249	38.84615385
Community blog	13	505	37
Educational IT course provider	2	49	24.5
Total	**35**	**1662**	**47.48571429**

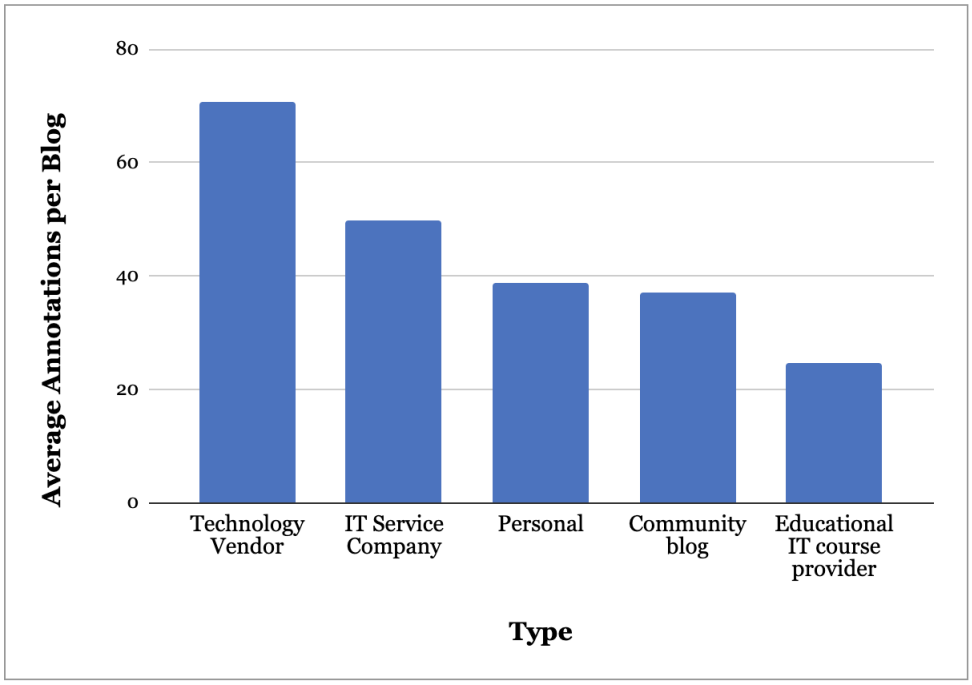


Figure 12: Number of AK concept annotations in each Blog Type

4.3.2 Topic vs AK concepts

Table 25 and Figure 13 shows the number of AK concepts per blog topic.

Table 25: Number of AK concept annotations in each Blog Topic

Topic	Count	Annotations	Annotations per Blog
List of alternative solutions	4	239	59.75
Comparison	12	598	52.90909091
Use Case	5	134	49.83333333
Solution Evaluation	11	582	36.33333333
How to	3	109	26.8
Total	**35**	**1662**	**47.48571429**

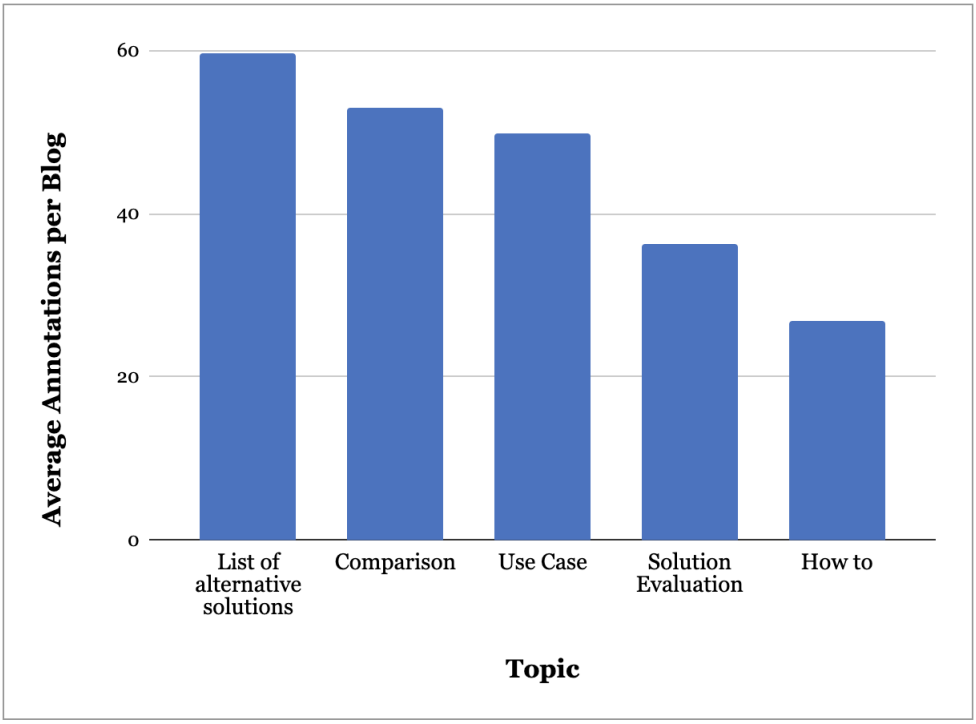


Figure 13: Number of AK concept annotations in each Blog Topic

4.3.3 Chi-square tests

Calculating the chi-squared tests for the co-occurrences of AK concepts per type and topic required 3 sets of tables to be created:

- Tables 26 and 27: Number of blogs in each type and concept
- Tables 28 and 29: Divide each cell by the total number of blogs in that type or topic
- Table 30 and 31: Calculate the chi-square values for each cell

Table 26: Number of Types of AK Concepts per Blog Type

AK Concepts	Total	Type				
		Community blog	Educational IT course provider	IT Service Company	Personal	Technology Vendor
ASTA	377	97	9	54	64	153
FEAT	291	63	7	40	46	135
REQ	144	44	2	16	17	65
CB	139	61	15	21	9	33
ADD	116	33	1	12	21	49
UR	91	19	7	13	16	36
DR	89	23	0	11	11	44
CONF	88	46	5	7	7	23
PAT	85	43	0	5	8	29
CASE	83	39	2	15	5	22
ALT	82	9	1	18	17	37
TRO	48	15	0	9	16	8
QEV	29	13	0	1	12	3

Table 27: Number and Types of AK Concepts per Blog Topic

AK Concepts	Total	Topic				
		Comparison	How to	List of alternative solutions	Solution Evaluation	Use Case
ASTA	377	144	14	52	141	26
FEAT	291	121	26	49	83	12
REQ	144	46	5	33	53	7
CB	139	36	21	4	64	14
ADD	116	47	3	28	31	7
UR	91	41	4	8	34	4
DR	89	32	3	26	23	5
CONF	88	20	6	4	40	18
PAT	85	12	1	9	54	9
CASE	83	21	14	5	25	18
ALT	82	29	8	20	17	8
TRO	48	25	4	1	13	5
QEV	29	24	0	0	4	1

Table 28: Values in Table 26 divided by number of blogs per type used in Step 3

AK Concepts	Type					Annotations per concept
	Community blog	Educational IT course provider	IT Service Company	Personal	Technology Vendor	
ASTA	7.461538462	4.5	9	12.8	17	**50.76153846**
FEAT	4.846153846	3.5	6.666666667	9.2	15	**39.21282051**
REQ	3.384615385	1	2.666666667	3.4	7.222222222	**17.67350427**
CB	4.692307692	7.5	3.5	1.8	3.666666667	**21.15897436**
ADD	2.538461538	0.5	2	4.2	5.444444444	**14.68290598**
UR	1.461538462	3.5	2.166666667	3.2	4	**14.32820513**
DR	1.769230769	0	1.833333333	2.2	4.888888889	**10.69145299**
CONF	3.538461538	2.5	1.166666667	1.4	2.555555556	**11.16068376**
PAT	3.307692308	0	0.8333333333	1.6	3.222222222	**8.963247863**
CASE	3	1	2.5	1	2.444444444	**9.944444444**
ALT	0.6923076923	0.5	3	3.4	4.111111111	**11.7034188**
TRO	1.153846154	0	1.5	3.2	0.8888888889	**6.742735043**
QEV	1	**0**	0.1666666667	2.4	0.3333333333	3.9
Annotations per Type	38.84615385	24.5	37	49.8	70.77777778	**220.9239316**

Table 29: Values in Table 27 divided by number of blogs per topic used in Step 3

AK Concepts	Topic					Annotations per concept
	Comparison	How to	List of alternative solutions	Solution Evaluation	Use Case	
ASTA	12	4.666666667	13	12.81818182	5.2	**47.68484848**
FEAT	10.08333333	8.666666667	12.25	7.545454545	2.4	**40.94545455**
REQ	3.833333333	1.666666667	8.25	4.818181818	1.4	**19.96818182**
CB	3	7	1	5.818181818	2.8	**19.61818182**
ADD	3.916666667	1	7	2.818181818	1.4	**16.13484848**
UR	3.416666667	1.333333333	2	3.090909091	0.8	**10.64090909**
DR	2.666666667	1	6.5	2.090909091	1	**13.25757576**
CONF	1.666666667	2	1	3.636363636	3.6	**11.9030303**
PAT	1	0.3333333333	2.25	4.909090909	1.8	**10.29242424**
CASE	1.75	4.666666667	1.25	2.272727273	3.6	**13.53939394**
ALT	2.416666667	2.666666667	5	1.545454545	1.6	**13.22878788**
TRO	2.083333333	1.333333333	0.25	1.181818182	1	**5.848484848**
QEV	2	0	0	0.3636363636	0.2	2.563636364
Annotations per Topic	49.83333333	36.33333333	59.75	52.90909091	26.8	**225.6257576**

Table 30: Chi-square test on AK concept per blog Type

AK Concepts	Type				
	Community blog	Educational IT course provider	IT Service Company	Personal	Technology Vendor
ASTA	0.16	0.1	0	0.11	0.01
FEAT	0.51	0.04	0.04	0	0.53
REQ	0.02	0.13	0.02	0	0.32
CB	0.08	11.48	0.08	1.83	1.64
ADD	0.1	0.29	0	0.06	0.02
UR	0.16	1.51	0.04	0.09	0
DR	0.1	0.47	0.15	0.05	0.42
CONF	0.75	0.56	0.03	0.2	0.12
PAT	1.22	0.29	0.02	0	0.01
CASE	0.41	0.17	0.08	0.33	0.03
ALT	0.47	0.08	0.19	0.04	0.01
TRO	0.23	0.1	0.02	1.22	0.42
QEV	0.06	0.01	0	1.56	0.21

Table 31: Chi-square test on AK concept per blog Topic

AK Concepts	Topic				
	Comparison	How to	List of alternative solutions	Solution Evaluation	Use Case
ASTA	0.14	1.24	0	0.19	0
FEAT	0.05	0.55	0.13	0.4	1.1
REQ	0	0.45	1.71	0.04	0.12
CB	0.23	4.61	3.92	0.16	0
ADD	0.01	0.6	1.7	0.08	0
UR	0.18	0.01	0.05	0.01	0
DR	0.03	0.24	2.55	0.12	0
CONF	0.11	0.11	1.24	0.06	2.41
PAT	0.35	0.51	0	2.26	0.01
CASE	0.25	2.29	1.36	0.07	1.67
ALT	0	0	0.41	0.5	0.17
TRO	0.09	0.02	0.58	0.09	0.06
QEV	2	0.02	0.06	0.15	0.59

5 Discussion

5.1 Discussion Step 1: Types

Often the blog type was selected based on the attributes of the company or website as a whole, rather than the specific web page. This resulted in other blog posts on the website often also needing to be examined.

The defining information for categorisation was commonly found at the bottom of the page, in the menu, or by clicking the logo in the top left of the page. At these locations, it was typical to find links to important pages titled *"about me, about us, about, about author, about [name of company], home, our story, who we are, what is [name of company], company"*.

The most important attributes to determine a blog type is the number of authors (often indicated by personal pronouns) and whether they are employed by the website company, and whether the company provides a product or service. Common questions that were asked during Step 1 to determine the blog types are:

1. Who can post on this blog? One person or multiple people? If multiple people, do they work for a company or can any person from the general public post to the blog?
2. Is this blog run by a company? If so, do they provide a product or service? Do users have to pay to receive any form of information (eg. to view articles, subscriptions, certifications)?

Figure 2 shows that most blog types have a a large range of relevance values and an average relevance ranking of around 3, which represents a Medium Relevance. This is understandable as it is the halfway point between 1 and 5. "Technology Vendor" and "IT Service Company" blogs have a very normal distribution. "Educational IT course providers" and "Magazines and Newspapers" are largely skewed left, meaning they generally have a higher amount of highly ranked blogs, although neither type contain any blogs of relevance 5.

The codes created in the column in the spreadsheet describing the reasons or attributes that influenced the categorisation were analysed. The following commonalities were recognised:

- Personal blogs often contain the words "I, my, me" and "About me", and often have the author's name in the URL or title and describes the qualifications or career of a single person.
- Technology Vendor blogs often contained menu items titled "products", "pricing" or "services", often mentioned prices, and used words "client, customers, services, supply, provide".
- Technology Specific Community blogs often have a URL containing the name of the brand/company that owns the mentioned specific technology.
- University websites often end in `.edu` or contain the word "university" in the URL or title.
- Blogs by an Educational IT Course Provider often contained the words "certification, courses, training, learn, platform", and always included prices, usually in the form of a list.

Table 8 and Figure 3 show that "Personal blogs" are the most relevant blog type. "Community blogs" are the most frequent and very relevant, whereas blogs by "Universities" and "Magazines and Newspapers" are the least frequent and least relevant. "Educational IT Course Providers" are highly relevant but not frequent, which can be explained as their primary goal is education and therefore they are very informative.

The correlation coefficient of 0.5326729 shows a moderately strong positive correlation between average relevance and number of blogs per type. There is a strong correlation between blog type and task.

5.2 Discussion Step 2: Topics

The most important indicator for blog topics are titles. Comparison blogs often contain the terms "alternative" and "vs" in the blog title.

Blogs with the topics "List of alternative solutions" and "Comparison" mention multiple technology solutions, while "Solution Evaluation" is specific to one solution.

The most interesting observation from Figure 6 is that "How To" blogs had the lowest frequency, however the highest relevance. A possible explanation for such a high relevance ranking is since "How To" blogs give very specific and basic information and instructions on how to implement a technology, these blogs possibly provided the answer to each task simply and easily. It is is worth noting that although these blogs could provide quick and basic information, they lack significant architectural knowledge, as seen in Table 25 from Step 3, showing "How To" blogs contain the least amount of AK concepts.

Other observations from Figure 6 are that "Comparison" and "List" blogs were the second and third highest respectively in both frequency and relevance. "Solution Evaluation" blogs were the most common, however had the second lowest relevance. "Use Case" blogs were low in both frequency and relevance.

There is no significant correlation between blog type and topic.

5.3 Discussion Step 3: AK concepts

1632 quotations and 1662 codes were annotated in 35 blogs in Step 3. The amount of Quotations and Codes are not equal since some sentences were coded with more than one AK concept, however this difference is very minimal. There are on average 47 AK concepts per blog. The average number of AK concepts in blogs with a relevance of 5 is 49.8, a relevance between 4 and 5 is 46.4, and a relevance of 4 is 45.8. Therefore it can be deduced that more AK concepts exist in blogs with a higher relevance ranking.

"Technology Benefits and Drawbacks", followed by "Technology Features", were the most common AK concepts mentioned in blogs. "Requirements and Constraints", "Component Behaviour", and "Recommended Design Decisions" were also common.

Table 26 and 27 shows that is it very common to have "Technology Benefits and Drawbacks" and "Technology Features" in blogs with the type "Technology Vendor" and topics "Comparison" and "Solution Evaluation". It is also common to have many "Technology Benefits and Drawbacks" AK concepts in blogs with the topic "Solution Evaluation". This makes a lot of sense since evaluations and comparisons often involve stating pros and cons.

The most common co-occurring AK concepts in blogs seen in Table 22 are "Technology Benefits and Drawbacks" and "Technology Features". This is understandable since the definition of "Technology Benefits and Drawbacks" (from Table 19) are the *"advantages and disadvantages of certain technology solutions or features."*. Usually both these AK concepts describe a characteristic of a technology, however they are differentiated based on whether this is stated in a subjective (Technology Benefits and Drawbacks) or objective (Technology Features) manner.

The second and third most commonly co-occurring AK concepts seen in Table 22 were "Decision Rules" with both "Requirements and Constraints" and "Recommended Design Decisions". This is also very understandable since the definition of "Decision Rules" (from Table 19) is a *"conditional recommendation"*, hence it provides "Recommended Design Decisions" often based on "Requirements and Constraints". Decision Rules are often in the form ``if [Requirements and Constraints] then [Recommended Design Decision]"..

Blog types "Technology Vendor" and "Community blog", and blog topics "Comparison" and "Solution Evaluation" contain the most AK concepts, as seen in Figures 12 and 13.

6 Conclusion

This study was performed with the goal of understanding what type, topics and AK concepts exist in architectural blogs. Based on this goal, I used Grounded Theory to determine definitions, examples, common indicators, frequency and relevance of architectural blog types and topics, and I used Qualitative Content Analysis to determine definitions, examples, common indicators and frequency of AK concepts in architectural blogs. In addition I performed correlation comparisons between blog types, tasks, topics and AK concepts, as well as analysed the distribution of types and topics within each AK concept and the co-occurrences of AK concepts.

926 architectural blogs were classified into types, 257 architectural blogs were classified into topics, and 1662 AK concepts were annotated from 35 architectural blogs.

6.1 Research Questions

The conclusions of the three Research Questions are as follows:

1. **RQ1: What are the types of architectural blogs?**

   There are 31 categories, grouped into 7 main categories of types of architectural blogs:

   - Community Blog
   - Technology Vendor
   - Personal blog
   - IT Service Company
   - Educational IT course provider
   - Magazines and Newspapers
   - University blog

   Blog types can commonly be determined by the number of authors and the company that runs the website. Relevance 3 is the most common relevance ranking for architectural blogs. `Number of blogs` and `Relevance per Type` are moderately strongly correlated. "Personal" blogs are the most relevant blogs but are not very frequent. "Community" blogs are the most frequent and are highly relevant. "University" blogs and "Magazines and Newspapers" are the least frequent and least relevant blogs.

2. **RQ2: What topics are discussed in architectural blogs?**

   There are 5 categories of topics of architectural blogs:

   - List of alternative solutions
   - Comparison
   - Solution Evaluation
   - Use-Case
   - How to

   Blog topics can commonly be determined through their title. "How To" blogs are the most relevant but least frequent blog. "List of alternative solutions" and "Comparison" blogs both have a medium frequency and relevance. "Solution Evaluation" blogs are the most frequent however have low relevance. "Use Case" blogs were low in both frequency and relevance.

3. **RQ3: What AK concepts are discussed in architectural blogs?**

   There are 13 AK concepts discussed in architectural blogs:

   - Architecture Configuration
   - Architecture Pattern
   - Component Behavior
   - Decision Rules
   - Quantitative evaluation
   - Recommended Design Decisions
   - Requirements and Constraints
   - Solution alternatives
   - Technology Benefits and Drawbacks
   - Technology Features
   - Technology Use-Cases
   - Trade-offs
   - User Request

   There are on average 47 AK concepts per architectural blog. Blogs with a higher relevance contain more AK concepts. "Technology Benefits and Drawbacks" and "Technology Features" are the most common AK concepts mentioned in blogs, most commonly occur together in a blog, and often occur in "Technology Vendor" (type), "Comparison" (topic), and "Solution Evaluation" (topic) blogs. "Decision Rules" often occur with both "Recommended Design Decisions" and "Requirements and Constraints". Blog types "Technology Vendor" and "Community" blog, and blog topics "Comparison" and "Solution Evaluation" contain the most AK concepts compared to other blog types and topics.

6.2 Threats to Validity

Remarks regarding threats to validity of this project are as follows:

- **Construct validity:** The source of the dataset used in this project, Soliman et al. [14], mitigated threats to construct validity by training the participants beforehand, randomly allocating the sequence of tasks, and using a plugin to capture the participants' input and store it in a database for analysis.
- **External validity:** The dataset of Step 1 accurately represented that of Soliman et al. [14] since every blog was analysed. The sample from Step 2 was calculated to be statistically significant, proportionate to the frequency and relevance results of Step 1, and randomly selected using a random number generator. The sample from Step 3 began at the highest relevance, since it contains the most AK concepts, and were sequentially selected in descending order until the goal of 1500 annotations was reached.
- **Reliability:** I attempted to ensure consistency during the classification process using Grounded Theory in Steps 1 and 2 by memoing, coding justifications for decisions, and creating tables of definitions, examples and attributes. During each step of this project, my supervisor randomly sampled blogs, examined and compared them to the category it was assigned, and a discussion was had. In situations where he felt my decisions were incorrect or needed improvement, we mutually came to conclusions and resolutions.

  One of my concerns during this project was too few re-iterations and discussions during the process of Step 3. Although I performed annotations to the best of my ability and personally feel I did sufficient background preparation for Step 3, it is a clear threat to reliability as there is a risk of human error and subjective conceptual understandings of AK concepts that may have influenced the results of Step 3. Thus if a similar task were to be completed in the future, ensuring correct understanding of AK concepts and expectations of the process of annotating using Atlas.ti, with sufficient discussions and revisions is needed.

6.3 Future Work

In terms of future work, it is possible to combine the results from this project with those of previous studies by Soliman et al., into a database. An application, repository or tools could be created for automatically mining, classifying, and capturing AK from web pages and support AK management systems. It could include clear indicators of which websites and blogs are in specific categories, and which AK concepts are in each web page. It could have a user interface for software engineers to easily find architectural knowledge to make accurate and informed design decisions for their specific needs.

A Appendix

Table A.1: List of URLs used in Step 3 for annotating AK concepts

Doc No.	URL
1	https://geekflare.com/best-stock-market-api/
2	https://bravenewgeek.com/tag/activemq/
3	https://www.openlogic.com/blog/activemq-vs-rabbitmq
4	https://medium.com/@rameez.s.shaikh/build-a-chat-application-using-spring-boot-websocket-rabbitmq
5	https://devopedia.org/cohesion-vs-coupling
6	https://blog.scottlogic.com/2018/07/06/comparing-streaming-frameworks-pt1.html
7	https://dzone.com/articles/develop-a-java-app-with-kafka
8	https://www.ibm.com/cloud/blog/rapidly-developing-applications-part-6-exposing-and-versioning-apis
9	https://wso2.com/whitepapers/microservices-in-practice-key-architectural-concepts-of-an-msa/
10	https://www.kai-waehner.de/kafka-versus-pulsar-event-streaming-comparison-features-myths-explored
11	http://ravindranaik.com/which-messaging-queue-is-best/
12	https://chatbotsjournal.com/why-knowledge-bases-and-chatbots-are-the-future-of-tech-support-790e238295cc
13	https://www.infoq.com/articles/AMQP-RabbitMQ/
14	https://freshcodeit.com/blog-introduction-to-message-brokers-part-1-apache-kafka-vs-rabbitmq
15	https://www.upsolver.com/blog/popular-stream-processing-frameworks-compared
16	https://stackabuse.com/reading-and-writing-json-in-java/
17	https://www.confluent.io/blog/build-deploy-scalable-machine-learning-production-apache-kafka/
18	https://dzone.com/articles/microservice-architecture-and-design-patterns-for
19	https://eclipsesource.com/blogs/2013/04/18/minimal-json-parser-for-java/
20	https://www.cloudamqp.com/blog/2019-12-12-when-to-use-rabbitmq-or-apache-kafka.html
21	https://www.infoq.com/articles/apache-kafka/
22	https://www.infoq.com/articles/microservices-intro/
23	https://programmer.help/blogs/performance-comparison-of-several-common-json-libraries-in-java.html
24	https://linuxhint.com/rabbitmq-vs-apache-kafka/
25	https://tanzu.vmware.com/content/blog/understanding-when-to-use-rabbitmq-or-apache-kafka
26	https://hevodata.com/learn/best-big-data-etl-tools/
27	https://blog.feathersjs.com/design-patterns-for-modern-web-apis-1f046635215
28	https://www.fasthosts.co.uk/blog/rabbitmq-and-message-brokers/
29	https://sloboda-studio.com/blog/how-to-use-nlp-for-building-a-chatbot/
30	https://www.janbasktraining.com/blog/what-is-flume/
31	https://www.edureka.co/blog/microservice-architecture/
32	https://datavirtuality.com/blog-etl-tools-and-processes/
33	https://www.astera.com/type/blog/rest-api-definition/
34	https://dzone.com/articles/apache-flume-and-data-pipelines
35	https://towardsdatascience.com/best-free-and-paid-stock-market-apis-for-2020-11adb98e7023

Blog Type		Average Relevance	Number of Blogs per Relevance Ranking					
Type	**Sub-Type**		**Total**	**Relevance 5**	**Relevance 4**	**Relevance 3**	**Relevance 2**	**Relevance 1**	**Relevance NA**
Personal blog		3.221088382	117	7	32	43	22	7	6
IT Service Company		3.094832251	88	5	18	34	25	4	2
Educational IT course provider		3.143613001	21	0	5	9	6	1	0
Magazines and Newspapers		2.458333333	18	0	1	5	9	1	2
University		2.65	4	0	0	3	0	1	0
Community blog	Community blog	3.186055556	279	12	77	105	51	20	14
	Community blog on a tutorial site	3.183333333	9	1	0	7	1	0	0
	Community blog on an Educational IT course provider site	3.5	4	0	2	2	0	0	0
	Technology specific community blog	2.895175439	40	3	9	10	11	5	2
	Community blog: total	3.154718426	332	16	88	124	63	25	16
Technology Vendor	Technology Vendor	2.916691729	98	5	19	32	33	6	3
	Technology Vendor: analytics	2.333333333	3	0	0	1	2	0	0
	Technology Vendor: API	3.4	5	0	3	1	1	0	0
	Technology Vendor: authentication	3.25	4	1	0	2	1	0	0
	Technology Vendor: books	3.204081633	7	0	3	3	0	1	0
	Technology Vendor: chatbots	2.25	4	0	0	2	1	1	0
	Technology Vendor: cloud	3.315756712	42	0	17	15	5	3	2
	Technology Vendor: eCommerce	2.633333333	5	0	0	2	2	1	0
	Technology Vendor: Finance	3.5	3	0	1	2	0	0	0
	Technology Vendor: healthcare	2.6	5	0	0	3	2	0	0
	Technology Vendor: integration	2.778571429	14	0	2	6	6	0	0
	Technology Vendor: total	2.995225313	190	6	45	69	53	12	5
Not a blog	Tutorial	3.287354514	36	0	14	11	8	1	2
	Forum	1.666666667	3	0	0	1	0	2	0
	Standards organization	2.5	2	0	0	1	1	0	0
	Scientific Journal publisher	1.5	2	0	0	0	1	1	0
	Book chapter	3.5	1	0	0	1	0	0	0
	Linkedin page	2	1	0	0	0	1	0	0
	Not a blog	2.8	5	0	2	1	1	1	0
	Not a blog: total	3.01689525	50	0	16	15	12	5	2
Not assigned	No Relevance	0	11	0	0	0	0	0	11
	Not assignable	2.922222222	17	0	6	3	3	3	2
	Relevance 1	1.034482759	58	0	0	0	0	58	0
	Page error	3.361666667	20	3	7	4	3	3	0
	Not assigned: total	1.668972747	106	3	13	7	6	64	13
TOTAL			**926**	**37**	**218**	**309**	**196**	**120**	**46**

Figure A.1: Number of Blogs and Average Relevance per Blog Type including sub-categories from Step 1.

References

[1] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice (2nd Edition)*. Addison-Wesley Professional, 2003.

[2] Manoj Bhat, Klym Shumaiev, Andreas Biesdorf, Uwe Hohenstein, and Florian Matthes. Automatic extraction of design decisions from issue management systems: A machine learning based approach. In *ECSA*, 2017.

[3] T. Bi, P. Liang, A. Tang, and X. Xia. Mining architecture tactics and quality attributes knowledge in stack overflow. *Journal of Systems and Software*, 180, October 2021.

[4] Humberto Cervantes and Rick Kazman. *Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering)*. 05 2016.

[5] L. Fu, P. Liang, X. Li, and C. Yang. Will data influence the experiment results?: A replication study of automatic identification of decisions. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 614–617, Los Alamitos, CA, USA, mar 2021. IEEE Computer Society.

[6] Ian Gorton, Ruochen Xu, Yiming Yang, Hanxiao Liu, and Guoqing Zheng. Experiments in curation: Towards machine-assisted construction of software architecture knowledge bases. *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 79–88, 2017.

[7] Anton Jansen and J. Bosch. Software architecture as a set of architectural design decisions. *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, pages 109–120, 2005.

[8] Philippe Kruchten, Patricia Lago, and Hans van Vliet. Building up and reasoning about architectural knowledge. In Christine Hofmeister, Ivica Crnkovic, and Ralf Reussner, editors, *Quality of Software Architectures*, pages 43–58, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[9] Philipp Mayring. *Qualitative content analysis - theoretical foundation, basic procedures and software solution*. 01 2014.

[10] M. Soliman, M. Riebisch, and U. Zdun. Enriching architecture knowledge with technology design decisions. In *2015 12th Working IEEE/IFIP Conference on Software Architecture*, pages 135–144, May 2015.

[11] Mohamed Soliman, Matthias Galster, and Paris Avgeriou. An exploratory study on architectural knowledge in issue tracking systems. *CoRR*, abs/2106.11140, 2021.

[12] Mohamed Soliman, Matthias Galster, and Matthias Riebisch. Developing an ontology for architecture knowledge from developer communities. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 89–92, 2017.

[13] Mohamed Soliman, Matthias Galster, Amr Rekaby Salama, and Matthias Riebisch. Architectural knowledge for technology decisions in developer communities: An exploratory study with stackoverflow. *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 128–133, 2016.

[14] Mohamed Soliman, Marion Wiese, Yikun Li, Matthias Riebisch, and Paris Avgeriou. Exploring web search engines to find architectural knowledge. *2021 IEEE 18th International Conference on Software Architecture (ICSA)*, pages 162–172, 2021.

[15] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. Grounded theory in software engineering research: A critical review and guidelines. In *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, page 120–131, New York, NY, USA, 2016. Association for Computing Machinery.