A Topic Modeling Approach to Categorizing API Customer Value Propositions

by

Mohamed Amin

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Technology Innovation Management

Carleton University
Ottawa, Ontario

Abstract

Companies increasingly use Application Programming Interfaces (APIs) to create new value propositions in the digital world. However, there is little knowledge about the type of value propositions that public APIs enable companies to provide. Thus, this study analyzes data from 13,829 API descriptions published in programmableweb.com to: (i) identify and categorize current and emerging API customer value propositions, (ii) analyze the evolution of API customer value propositions, and (iii) showcase Topic Modeling as a feasible technique to study API customer value propositions. This research contributes towards API service systems research by: (i) identifying six categories of API customer value propositions, and (ii) revealing a long tail distribution of API customer value propositions that is flattening.

Acknowledgements

I would like to start by thanking my Supervisor, Professor Mika Westerlund, for his guidance, tolerance, and open mind for new ideas and methods. I would also like to thank the TIM faculty: Professor Tony Bailetti, for teaching me what a Customer Value Proposition in the real world means, Professor Michael Weiss, for giving me the fundamental idea behind this research, and Professor Steven Muegge, for teaching me meticulousness in research and writing. Although I may have not met their standards yet, I know I am headed in the right direction.

I would like to thank my parents for their support and encouragement throughout my TIM journey and their willingness to do everything to help me complete this degree. I would also like to thank my in-laws for the numerous times they had to babysit my kids as my wife and I are engulfed in grad school.

Super special thanks, appreciation and love to my wife, Maha, for everything she did to help me complete this degree. This includes, but by no means limited to: taking full care of every aspect of the house, bearing the number of nights I did not make it home for dinner, patiently responding to my complaints about research methods, and literature reviews, her unconditional support, and most importantly for saying “NO” every single time I mentioned I am going to quit the program.

Table of Contents

List of figures

List of tables

List of Notations

API	Application Programming Interface
CTM	Correlated Topic Models
DTM	Dynamic Topic Models
HTTP	Hyper Text Transfer Protocol
JSON	JavaScript Object Notation
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
PLSA	Probabilistic Latent Semantic Analysis
PLSI	Probabilistic Latent Semantic Indexing
RAML	RESTful API Modeling Language
RDF	Resource Description Framework
REST	Representational State Transfer
RPC	Remote Procedural Call
RSS	Rich Site Summary
RTM	Relational Topic Models
sLDA	Supervised Latent Dirichlet Allocation

SOAP	Simple Object Access Protocol
URL	Uniform Resource Locator
WADL	Web Application Description Language
WSDL	Web Service Definition Language
XML	Extensible Markup Language

1 Introduction

API service systems are collections of APIs – technical interfaces that provide programmatic access to a company's data or service – whose combination and procurement by API consumers result in novel applications such as Mashups (Barros & Dumas, 2006). Directories such as programmableweb.com represent service systems by aggregating, presenting, and allowing the discovery of collections of APIs and Mashups. Today, programmableweb.com is the largest repository of APIs and reports having 15,816 APIs in its directory. This research uses API service systems and programmableweb.com interchangeably.

The study of the API service systems is of high value (Lyu et al., 2014). Several studies have taken an empirical look at API service systems, in particular programmableweb.com, using network science techniques (Endres-Niggemeyer, 2013; Huang et al., 2012; Lyu et al., 2014; Yu & Woodard, 2008; Weiss & Gangadharan, 2010; Wiess & Sari, 2010; Weiss et al., 2013; Rosenblum, 2011). These studies use the links between Mashups and APIs as a focal point for their study.

In 2010, the number of Mashups was reported to be 5,028 and the number of APIs was reported to be 2,084 (Weiss & Sari, 2010). In 2016, the number of Mashups reported by programmableweb.com is 7,827, and the number of APIs reported is 15,816. These numbers show that while Mashup listings have seen a slow growth over the past six years, API listings have skyrocketed indicating that APIs are still ripe for exponential growth. This calls for the need of using different techniques and methods to study APIs to reveal new insights.

Researchers are urged to use practical approaches to solve real-life problems in the API economy (Tan et al., 2016). One practical problem that API service providers frequently face is figuring out how to create and communicate value using their APIs (Woods et al., 2011). Thus, customer value propositions are an important aspect of a company's API strategy, and little is known about this area.

Customer value propositions can be conceptualized as a communication practice (Ballantyne et al., 2011). In programmableweb.com, API providers use API descriptions to communicate customer value propositions to potential API consumers. By analyzing API descriptions, API customer value propositions can be identified and categorized.

This study uses Topic Modeling as a text mining technique to study how API providers present their customer value propositions to API consumers in programmableweb.com. Topic Modeling has emerged as a strong natural language processing technique for characterizing "topics" in a large corpus of text (Blei, 2012). By applying Topic Modeling to a dataset of API descriptions from programmableweb.com, we can empirically identify and categorize API provider's representation of customer value propositions and how these representations evolved over time revealing new insights and managerial implications.

1.1 Objective

The objective of this research is to apply Topic Modeling on a corpus of API descriptions from programmableweb.com to identify and categorize API customer value propositions.

1.2 Deliverables

This thesis has 5 deliverables.

- A categorization model of API customer value propositions.
- Topic trend graphs showing how topics evolved over the period from 2006 to 2016 revealing a long tail distribution that is flattening out.
- A showcase for using Topic Modeling to study API service systems.
- Recommendations for how API service providers, API service systems, and entrepreneurs can fill the industry gap.

1.3 Contributions

This research makes two types of contributions, contributions to the general body of scholarship and contributions to practice.

1.3.1 Contributions to scholarship

This research contributes towards scholarly knowledge by:

- Applying Topic Modeling as a research approach to studying service systems.
- Serving as a basis for further developing customer value propositions research in service systems by creating value maps and value networks.
- Revealing insights about the evolution of customer value propositions in service systems.

1.3.2 Contributions to practice

This research contributes towards practice by:

- The identification of an industry gap, where API service systems are falling short from acting as filters to connect demand and supply in the API economy.
- Allowing API service providers to categorize their APIs based on the new categorization model.

1.4 Relevance

The deliverables of this research would:

- Allow entrepreneurs and existing API providers to differentiate themselves by better articulating their customer value propositions
- Provide API service systems with a tool that allows them to study how value is created in their service systems and where the gaps are
- Provide API service systems with guidance as to how they can become filters closing the gap between supply and demand in the API service system.
- Provide researchers with an example of how Topic Modeling can be applied to study service systems.

1.5 Overview of the method and expected results

This is an inductive theory building research where descriptive theory is the ultimate goal of the study (Carlile & Christensen, 2004). Table 1 summarizes the steps taken during this study.

Step	Activity
Identifying constructs and	Identify constructs of customer value propositions

theoretical framework	and build a theoretical framework for data analysis
Data collection	Obtain descriptions of 13,829 APIs from www.programmableweb.com
Data preprocessing	Preprocessing data to prepare it for topic modeling by removing stop words and importing into Mallet (Topic Modeling and Mallet will be discussed in detail in chapter 3)
Topic Model training	Use Mallet's Latent Dirichlet Allocation (LDA) algorithm to train different Topic Models while increasing the number of topics in every training
Evaluation of models	Evaluate the trained models based on a set of criteria to find the model with the optimal number of topics for this study.
Interpretation of results	Interpret results in light of identified constructs
Categorization	Categorize the results based on the initial theoretical framework. Produce the final categorization model.
Topic Trend Analysis	Produce topic trend graphs showing the evolution of topic representations over the period from June 2005 to July 2016

Categorization Model and Statements of Association	Produce final categorization model and statements of association by comparing with extant literature.

Table 1 - Summary of Research Method

1.6 Organization of the thesis

This research is organized into six chapters. After this introduction, chapter 2 reviews the literature streams: (i) Web service composition, (ii) API service systems, and (iii) API customer value propositions. Chapter 3 details the research method used. Chapter 4 provides the results and chapter 5 discusses the results of this research. Chapter 6 concludes the study and provides research limitations and suggestions for future research.

2 Literature Review

The literature review is organized as three streams. The first stream covers web services composition. The second stream surveys API service systems. The third stream reviews API customer value propositions. This chapter concludes with a summary and synthesis of the lessons pertinent to this research.

2.1 Web service composition

Service composition is the process of building new applications and services (composites) out of existing web services (Singh, 2001). Its philosophy of black box integration made web service composition the most prominent way for B2B integration (Garriga et al., 2016). Service composition literature can be organized to answer three questions:

- **What is composed?** Literature in this category describe the types of components that are being composed by web services, the interaction protocols that web services use, the description languages of the components and the format of data exchange, and finally the target applications or the final product of the composition.
- **How is it composed?** Literature in this category describe the process of service selection, service discovery, and service recommendation.
- **Who is performing the composition?** Literature in this category describe the types of developers that perform the composition.

In the following, the present study discusses each of these areas in more detail.

2.1.1 What is composed?

2.1.1.1 Component types

The type of component used in the composition can be broken down into three categories: (i) Data component encapsulates a web resources using a certain format and a defined data structure; (ii) Application logic components provide business functionality such as checking stock availability or sending a text message; and (iii) User interface components are UI widgets such as login widgets, map widgets, and search widgets that require widget containers or engines to be instantiated (Lemos et al., 2015).

2.1.1.2 Interaction protocols

Components can be exposed in two ways: SOAP and REST. SOAP is an XML based protocol where information can be exchanged over HTTP or RPC (Box et al., 2000). REST is an architectural pattern that allows data to be represented as resources and accessed through dedicated URLs over HTTP (Fielding, 2000). REST is more a design philosophy based on navigational style while SOAP's design style is procedural. REST and SOAP are not necessarily opposite ways of doing the same thing, although debates in literature make them seem so (Muehlen et al., 2005). SOAP-based web services worked well inside enterprises while REST-based web services have taken momentum with Internet-scale applications (Lanthaler & Gutl, 2010). REST-based web services still face problems with formal description, discovery, and orchestration due to the lack of interface standards and universal service registries as in SOAP-based services (Lanthaler & Gutl, 2010).

2.1.1.3 Description languages and data formats

Components can be described in many different languages. The description languages are usually tied to interaction protocols. SOAP based web services can be described using Web Service Description Language (WSDL) (Christensen et al., 2001). REST based web services can be described in a variety of ways such as WADL (Hadley, 2006), RAML, and Swagger (Lemos et al., 2015). Data is exchanged between the service provider and the service consumer as JSON and RSS/Atom (Lemos et al., 2015). JSON is a text-based, human-readable, light weight data exchange format (Crockford, 2006). RSS (RSS Advisory Board, 2009) and Atom (Notingham & Sayre, 2005) are mainly used for web feeds such as blog posts and online news (Lemos et al., 2015) and are supported by Mashup tools (Beletski, 2008).

2.1.1.4 Target applications

Service compositions provide value for three types of target applications: (i) Mashups are applications that are composed of multiple data sources, (ii) Business Process consists of a set of activities, functional roles, and relationships that achieve business goals by providing a business function, and (iii) Scientific Workflows are reusable experiments and data pipelines developed and used by the scientific communities (Lemos et al., 2015).

2.1.2 How is the composition performed?

Selection is the process by which a web service is discovered, identified and used based on the given composition requirements. Static composition is when selection takes

place at design time when the developer is still building the composite service. Dynamic composition on the other hand involves selection happening at either deployment time when the composition is installed for execution or runtime when the composition is executed (Lemos et al., 2015).

Knowledge reuse is the underpinning nature of service composition. Reuse can be achieved by ways such as searching for and discovering an artifact based on a set of requirements, copying a composition from one repository or tool and pasting it into another, creating a replica of a composition by cloning it, and recommending artifacts that facilitate the composition process (Lemos et al., 2015).

Service discovery is one of the sub tasks of Service Oriented Architecture (Bachlechner et al., 2006). The lack of machine processable descriptions makes finding, understanding, and consuming web APIs a manually intensive process. (Maleshkova et al., 2010). Most of the web services discovery approaches are based on keyword search and category browsing with Google providing the best coverage and precision for finding web services (Bachlechner et al., 2006). Discovering and composing RESTful services requires a human manually searching through a web site such as Programmableweb.com that collects and categorizes these services (Lanthaler & Gutl, 2010).

Deeper understanding of how web APIs are developed, exposed, and described needs to be reached before suggesting major improvements in existing web API technologies (Maleshkova et al., 2010). Developers face difficulty in reusing web services in 3rd party

application due to the different standards used in exposing web services. This difficulty is hindering the full potential of web services. This difficulty can be addressed by the representation of services using RDF graphs in a Service Graph Base (Chen et al., 2012). This representation allows services to be queried regardless of their type using common SPARQL queries. Many more selection, discovery, and recommendation solutions were proposed in the literature. Below are some of the proposed solutions:

- Driven by the need for e-business community to automate management of supply chain, the discovery of web services based on capabilities they provide become critical (Paolucci et al., 2002)
- A technique for composing services based on Quality of Service using genetic algorithms (Canfora et al., 2005)
- A method for the dynamic composition and management of composite e-services that are built on top of other basic services (Casati et al., 2000)
- A system for assisting mashup developers by suggesting relevant outputs by calculating the conditional probability that the output will be included given the current state of the mashup (Elmeleegy et al., 2008)
- A mashup component recommendation mechanism based on a top-k ranking algorithm that exploits the shared characteristics between mashups to rank and recommend mashup components (Elmeleegy et al., 2008)
- An interactive method of recommending services based on reusable composition patterns to address the difficulty of developing mashups faced by mashup developers (Chowdhury et al., 2011)

- VisComplete: a tool that aids domain scientists with building scientific workflows and data pipelines by suggesting components for new pipelines and scientific workflows while being constructed using correspondence between pipelines in an existing database. (Koop et al., 2008)
- A dynamic recommendation system based on user feedback and collaborative filtering for dynamic web service selection using semantic matching service requirements (Manikrao & Prabhakar, 2005)
- A solution for the semantic matching of web services based on the capabilities they provide (Paolucci et al., 2002)
- A method for allowing software developers to annotate web services with semantic descriptions (Rajasekaran et al., 2005)

2.1.3 Who is performing the composition?

There are three types of users that are mainly targeted by the practice of service compositions: (i) Professional programmers are highly skilled developers who can build complex compositions and even new APIs and components, (ii) End-user programmers can glue together different components and APIs and create new applications. They understand the principles of service composition, but are not skilled enough to produce APIs and components. They are usually domain experts and most of their applications are context specific or for personal use, (iii) End-user app remixers are not skilled in software development at all, but understand the web and can stitch together rule based applications based on the notion of “If This Then That”. They usually use services such as

Zapier[1] and IFTTT[2] (Lemos et al., 2015).

2.2 API service systems

Web services are modular applications that can be invoked across the Web (Bai et al., 2009). The interface these applications expose to be invoked is an API. The Software Engineering Body of Knowledge defines an API as "a set of signatures that are exported and available to the users of a library or a framework to write their applications" (Bourque & Fairley, 2014:3-8). Web service systems consists of web services, service providers, service consumers, and service based applications (Lyu et al., 2014). The literature does not make a clear distinction between web service systems and API service systems. For the purposes of this study API service systems will be used to refer to service systems that enable API providers to publish, promote, and provision their APIs and API consumers to discover, select, and consume APIs (Wittern et al., 2014) and API service systems are actual instances of directories and repositories such as programmableweb.com that enable service systems.

The study of API service systems is of high value (Lyu et al., 2014). Information about APIs such as endpoint descriptions, data formats, and protocols provided by API providers play a key role in the consumption of APIs by API consumers. In addition, such information allows the evolution of the API service system by (Wittern et al., 2014):

- Enabling API consumers to easily discover and consume APIs by knowing about

[1] https://zapier.com/
[2] https://ifttt.com/

valuable API combinations and getting API suggestions based on requirements

- Enabling API providers to know about how their APIs are used, what needs are not met in the service system, and how they compare to competing APIs
- Enabling API service systems to know about gaps in the service system that need to be filled, how demand is changing in the service system, and what tooling is required for the service system to prosper

Several studies have empirically examined programmableweb.com. Most of these studies use the links between APIs and Mashups to construct service networks that can then be examined by social network analysis techniques. Methodologies for studying programmableweb.com were suggested based on network analysis techniques (Huang et al., 2012), phylogenetic trees (Weiss et al., 2013), and graph theory (Pan et al., 2012; Wittern et al., 2014).

Studies on programmableweb.com can be broken down into three groups: (i) studies that look at the structure of the service system and draw insights from visualizing it, (ii) studies that take an evolutionary look at the service system and reveal insight about its growth, (iii) studies that use empirical evidence to suggest recommendation and discovery solutions in the API service system.

Through studying the structure of programmableweb.com evidence showed that APIs are organized into three triers with Google maps at the center. The second and third tier are comprised of platform like APIs, while some of the second tier APIs and the third tier APIs act as data sources. Mashups are usually composed by combining APIs across the

three tiers (Yu & Woodard, 2008). The overall structure of the service system revealed a power-law distribution among Mashups to APIs, characteristic of a long tail distribution where a small number of APIs are used by the vast majority of Mashups and the remaining number of APIs are used by few or no Mashups (Yu & Woodard, 2008; Weiss & Gangadharan, 2010; Woodard, 2009). A power-law distribution occurs when a small number of occurrences account for the majority of observations and a large number of occurrences account for the rest of the observations (Clauset et al., 2009). The resulting curve has been described as the long tail curve (Anderson, 2006).

Programmableweb.com can be visualized at three levels, the web API graph, the tag graph, and the domain graph. This breakdown can aid discovery the service system (Lyu et al., 2014). Integration patterns were studied using social tags to reveal that mashups in the API service system experience hybrid integration patterns by integrating real-life applications in addition to composing APIs (Han et al., 2014). More recently, network analysis and visualization revealed that although APIs are proliferating in numbers, the service system is still seeing a power law distribution among API uses (Evans & Basole, 2016).

The evolution of the mashup service system can be attributed to niche formation around central hub APIs (Weiss et al., 2013). In studying the characteristics of developers in the API service system, evidence suggest that frequently used APIs in the API service system attract large amounts of users who have similar ideas and similar behavior. Thus, developers tend to use similar design patterns in creating mashups (Wang et al., 2009). There is evidence to suggest that copying among developers play a

significant role in the evolution of the mashup service system. This challenges the idea that selecting APIs is based on the APIs popularity (Weiss & Sari, 2010). The relationship between APIs in the API service system is one of complementarity and is affected by the position of the API in the service system. 'Powerful hub' APIs tend to attract niche data providers as complements and the chance two APIs will be used together increases as they are used together more often (Weiss & Gangadharan, 2010).

The idea that previous examples of how APIs were previously used influence how they will be used in the future could be used to recommend APIs to developers. By leveraging social information in the API service system, APIs could be re-ranked based on their history enhancing their discoverability (Torres et al., 2011). Detecting services based on their community structure is important for service computing. Two methods were suggested to detect communities in the API service system: competition-oriented based on the functional semantics of the web service, and collaboration-oriented based on topological analysis (Han et al., 2013). Link analysis and network prediction techniques were used to provide recommendations in the API service system (Huang et al., 2014). Finally, a time aware linear model was proposed to predict API popularity using the time series feature of APIs such as provider ranking and description (Wan et al., 2015)

2.3 API customer value propositions

This section reviews literature on customer value propositions in the API service system by first reviewing the notion of value propositions and its relation to service systems and service service system, and then reviews the general business value that APIs bring to

enterprises and value networks.

2.3.1 Value propositions and service systems

API service systems can be thought of as service value networks (Weinhardt et al., 2011). Networks play a central role in value creation and exchange that underpins Service Dominant (S-D) logic (Lusch & Vargo, 2006). Service system are comprised of complex combinations of goods, money, activities, and institutions where skills and knowledge based competencies are applied for the benefit of one another (Vargo & Lusch, 2008). S-D logic forms the basic underpinning of service system, and it refers to an evolving view of service provisions as the fundamental element of economic exchange in markets. In S-D logic, enterprises can only make value propositions and value creation is only achieved when the service is consumed (Vargo & Lusch, 2004).

Three service constructs make up value propositions in service systems: value propositions as invitations of service engagement between actors, engagement as alignment of connections and dispositions, and service experience as many-to-many engagement (Chandler & Lusch, 2015). Value propositions as seen through the S-D logic are understood to be reciprocal promises of value. In light of this understanding, value propositions are used as a communication practice that enhances the firm's engagement with suppliers, customers, and other beneficiaries (Ballantyne et al., 2011).

Value propositions consist of provision practices, representational practices, and management and organizational practices that integrate operant and operand resources into new value propositions through an innovation process (Skalen et al., 2014).

Customer value propositions communicate to target customer what matters most to them in terms of points of difference and points of parity compared with the next best alternative (Anderson et al., 2006).

In the context of service systems, value propositions play a key role in shaping the service system by acting as mechanisms for balancing resource sharing. In the service system, value propositions are a result of the actors' willingness to share and contribute their resources to others (Frow et al., 2014). They act as integrators and connectors between its actors, thus their change and evolution over time influence the shape of the service system and contribute to its health (Frow et al., 2014).

2.3.2 Business value of APIs

The benefits of adopting APIs for enterprises are: more agility to make firms more adaptable to changing requirements, less complexity in IT infrastructure to make it easier to deal with, increased reusability as software components can be reused to achieve different goals without the need of re-implementing them, and better interoperability as software components can interact and exchange information over standard interfaces (Hau et al., 2008) The adoption of APIs has led to increased performance in supply chain by reducing the negative effects of the complexity of information sharing (Kumar et al., 2007). APIs contribute to business process quality by complementing business process management in standardization, consolidation, and B2B integration (Beimborn & Joachim, 2011). APIs attempt to facilitate the job of the information worker by automating tasks such as searching, analyzing, reformatting, and

consolidating information (Lanthaler & Gutl, 2010). In general, service based software is complementary to component based software, and is suitable for systems with frequently changing requirements (Elfatatry, 2007).

Outside the walls of the enterprise, APIs are becoming increasingly important for companies as they allow them to give their partners and customers access to their services and assets (Wittern et al., 2014). Web APIs allowed developers to combine different data sources into Mashups. These Mashups in turn allow non-technical users to innovate quickly by easily accessing data and mixing it together to try new ideas and concepts (Hinchcliffe & Benson, 2011).

While service orientation facilitates integration of business components, businesses can operate in value nets using componentization (Cherbakov et al., 2005). A value network can be mapped using three elements: roles are played people or organizations in the network who perform the functions necessary for value creation, transactions are activities that originate from one role to another, and deliverables are the actual benefits that are moved during a transaction between two players (Allee, 2008).

Although the business value of APIs are well known within the walls of the enterprise, significant adoption of service technologies outside the enterprise environment has been slow. This is attributed to the lack of platforms that are able to deal with the heterogeneity in the service system and can provide simple yet powerful discovery mechanisms (Pedrinaci et al., 2010)

2.4 Summary and synthesis of key findings from the literature

Literature around the web service system can be conceptualized at three levels of abstraction (see Figure 1 for a representation of the levels). The highest level is literature on web service composition. Web service composition can be thought of as an abstract technical architecture. This body of literature discusses topics such as components, compositions, composers, and discovery. The second level focuses on one of the ways a web service composition can be implemented which is API service systems. This body of literature talks about service providers, service consumers, applications, and matching. At the third and most concrete level is literature that studied programmableweb.com. Programmableweb.com is one actual instance of the web service systems where there are API and Mashup listings.

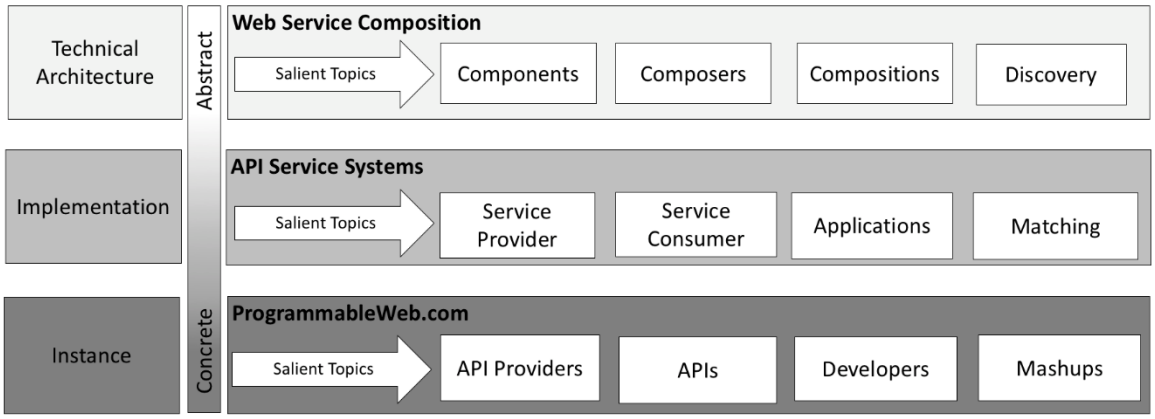

Figure 1 - Breakdown of literature

APIs are major drivers of value within the enterprise. They achieve technical and business value by streamlining processes, allowing the enterprise to be scalable and agile, and facilitating B2B integration. However, APIs are increasingly having trouble reaching their full potential outside the walls of the enterprise at the scale of the internet.

API service systems have emerged in recent years with a huge potential for achieving value at an internet scale. Empirically studying these service systems is extremely important to allow service providers, service consumers, and service systems to reach their full potential. Past empirical studies of the web service system focused on network science perspectives and revealed many insights into the formation of the service system, the key players, and how the service system evolves. However, empirical studies of topics that require semantic analysis such as customer value propositions in the service system are rare and face the difficulty of massive amounts of data that are impossible for humans to process manually. This is a gap in literature that needs to be filled.

3 Methods

This chapter discusses the research methods used in this thesis. An overview of Topic Modeling is given followed by details of the research steps.

3.1 Research approach

This is an inductive theory building research where descriptive theory is the ultimate goal of the study (Carlile & Christensen, 2005). The steps of this research are based on the three stages of theory building pyramid suggested by Carlile and Christensen (2005). This research uses Topic Modeling as a data analysis and text mining technique.

3.2 Unit of analysis

The unit of analysis is a corpus of API descriptions obtained from www.programmableweb.com. API descriptions are statements written by the API provider that describes to potential API consumers what the API does. Figure 2 shows a typical API description from programmableweb.com.

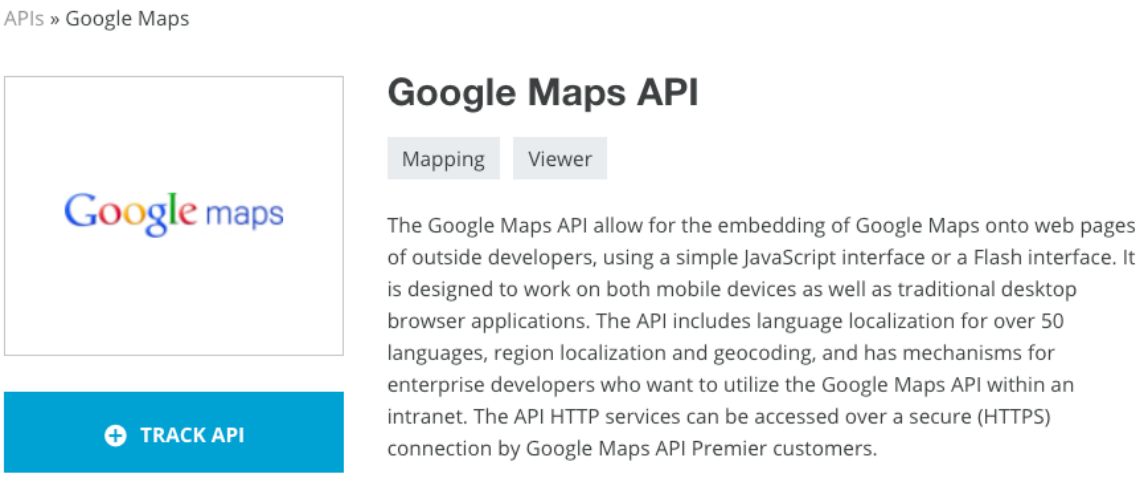

Figure 2 - An example of API description from programmableweb.com

3.3 Study period

The collected data covers the period from June 2005 to July 2016 and includes 13,829 API descriptions, API names, API dates of submission, and API tags. Programmableweb.com reports 15,816 APIs in its database. Hence, the data collected represents 87.4% of the total number of APIs published as of September 2016.

3.4 Topic Modeling

3.4.1 Probabilistic Topic Modeling

Probabilistic Topic Modeling is a set of statistical techniques and algorithms that allow thematic discovery and annotation of information in a large archive of documents (Blei, 2012). The techniques are based on the general field of Probabilistic Modeling. The three key elements of a topic model are a document, a topic, and a word. A document is a mixture of topics (Blei et al., 2003; Griffiths, & Steyvers, 2002; 2003; 2004; Hofmann, 1999; 2001). A topic is a theme discussed in one or more documents and is represented as a probability distribution over words (Steyvers & Griffiths, 2010).

The basic idea underlying probabilistic topic modeling is the formation of documents using a generative process. Documents are generated by choosing a probabilistic distribution over a number of topics. Then for each word in the document, a random topic is chosen according to this distribution and a word is drawn from the topic. The goal of topic modeling is to use standard statistical techniques to invert this process so that the set of topics that were responsible for the generation of the document can be deduced.

For example, let us assume that we want to generate this thesis using the probabilistic generative process. We will first go through a generative process to that will create a thesis and then describe how topics can be inferred from a given thesis by inverting the generative process.

The first step of the generative process is to pick a probabilistic distribution over a number of topics. To demonstrate, let us pick five topics and choose a probability distribution over them.

The five topics are:

Topic 1: Web Services	P(1) = 0.20
Topic 2: Service Composition	P(2) = 0.15
Topic 3: Value Propositions	P(3) = 0.13
Topic 4: Service systems	P(4) = 0.20
Topic 5: Topic Modeling	P(5) = 0.32

Step two would be to assign a random probability distribution for words in the above topics over a fixed vocabulary. This simply means that if we hypothetically have 1,000 distinct words that we can choose from to make up our documents, each topic would have a different probability distribution over this set of distinct words. For the sake of this example, we will show five words and their probability distributions, and represent the rest with '…'.

Topic 1: Web Services

Provider	0.04
Consumer	0.03
Service	0.07

HTTP	0.02
REST	0.09
…	

Topic 2: Service Composition

Selection	0.03
Discovery	0.04
Composition	0.08
Recommendation	0.05
Automation	0.07
…	

Topic 3: Value Propositions

Difference	0.02
Points	0.06
Customer	0.08
Competition	0.03
Value	0.09
…	

Topic 4: Service system

Mashup	0.05
API	0.02
Service	0.09
Growth	0.03
Network	0.08
…	

Topic 5: Topic Modeling

Word	0.04

Vocabulary	0.02
Document	0.08
Topic	0.09
LDA	0.05
…	

Step 3 in the generative process would be to generate each word in the document in two steps:

1- Randomly choose a topic given the distribution of topics above

2- Randomly choose a word from the corresponding distribution of the vocabulary

Due to that writing a thesis using this method would be overly exhaustive, we can reverse this process using standard statistical techniques to arrive at the topics and their corresponding distributions from a properly constructed thesis document. There are various standard statistical techniques that can do this job. Some of them include Collapsed Gibbs Sampling (Griffiths & Steyvers, 2002), Mean Field Variational Methods (Blei et al., 2003), Expectation propagation (Minka & Lafferty, 2002), and Collapsed Variational Inference (Teh et al., 2006). The algorithms that implement those techniques are the ones generally referred to as Topic Modeling algorithms.

The basic intuition behind these techniques is hiding (hence latent) the topics, the per-document topic distributions, and the per-document per-word topic assignments, while revealing the document only. Using an iterative process of sampling try to guess the topic and word distributions while conditionally given the current posterior distribution of the topics and words. After enough iterations topics are inferred (Blei, 2012; Steyvers

& Griffiths, 2010).

3.4.2 Types of Topic Modeling

Topic modeling algorithms evolved over the past two decades and saw many applications within areas such as digital humanities and journalism. Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) kick-started Topic Modeling by proposing an automatic information indexing and retrieval method based on Singular Value Decomposition (Deerwester et al., 1990). LSA solved a problem that existed in traditional information retrieval methods where the synonymy and polysemy of words were not accounted for. Synonymy meant users referring to the same object using different ways and words, and polysemy meant that the same word can have more than one meaning (Deerwester et al., 1990). Probabilistic Latent Semantic Analysis/Indexing (PLSA or PLSI) based on the likelihood principle, added the statistical foundation that LSA missed by defining a probabilistic generative model (Hofmann, 1999).

LSA and PLSA were based on three principles: deriving semantic information from word-document co-occurrence matrix; dimensionality reduction; and presenting words and documents as points in Euclidean space (Steyvers & Griffiths, 2010). The third principle makes the assumption of exchangeability; the order of words and documents is not important.

Latent Dirichlet Allocation(LDA) uses LSA and PLSA's first two principles but expresses words and documents in terms of probabilistic topics rather than points in Euclidean space (Blei et al. 2003; Steyvers and Griffiths, 2010). LSA, PLSA, and LDA are static topic

models because they assume that content does not change over time. Dynamic Topic Models (DTM) capture the evolution of topics in a sequentially ordered set of documents over time. Dynamic Topic Models use Gaussian models to incorporate time dynamics in the topic model (Blei & Lafferty, 2006a) while Correlated Topic Models (CTM) represent correlation using logistic normal distribution for approximate posterior inference where documents cannot be correlated (Blei & Lafferty, 2006b).

The evolution of topic models continues allowing features other than words to be included in the model. Supervised Latent Dirichlet Allocation (sLDA) accommodates external responses pairing them with documents for the goal of using the model to predict the responses. sLDA moves topic modeling into the supervised machine learning space where predictive analytics is the ultimate goal rather than the unsupervised machine learning space where clustering is the ultimate goal (Blei & Mcauliffe, 2008).

Relational Topic Models (RTM) is another supervised topic model where the external responses are specifically links between the documents. Using links and words the model can be used to map out the document network, predict links between them given words, and predict words in them given links (Chang & Blei, 2009).

Hierarchical Dirichlet Process (HDP) on the other hand extends LDA into groups of mixture models that are themselves linked by a mixture model. This allows cases where multiple corpora are to be analyzed together where topics need to be identified within each corpus as well as among the set of corpora (Teh et al., 2006)

3.4.3 Applications of Topic Modeling

Topic Modeling has been used in many applications such as analyzing scholarly publications, data journalism and journalism research, analyzing web content, sentiment analysis, and analyzing software source code. Table 2 shows an account of Topic Modeling applications in various domains.

Domain	**Studies**
Scholarly Publications	• Citation influence (Dietz et al., 2007) • Scientific publications (Erosheva et al., 2004) • Measuring Scholarly impact (Gerrish & Blei. 2010) • Studying the history of ideas (Hall et al., 2008) • Construct identity in literature reviews (Larsen & Bong, 2016) • Recommending scientific articles (Wang & Blei, 2011)
News and Journalism	• Analyzing news articles (Newman & Chemudugunta, 2006) • Comparing twitter and traditional media (Zhao et al., 2011) • Analysis of multiple online news sources (Chuang et al., 2014) • Analyzing journalistic text (Jacobi et al., 2015) • Analyzing Wikipedia (Ni et al., 2009) • Event summarization from news and social media (Geo et al., 2012)
Author Analysis	• Finding influential authors in a digital library (Mimno & McCallum, 2007) • Author centric ranking of web content (Ali & LaPaugh, 2013) • Predicting author attributes from Twitter messages (Mccollister et al., 2015)
Web Content and Sentiment Analysis	• Opinion integration (Lu & Zhai, 2008) • Sentiment analysis (Lin & He, 2009) • Sentiment analysis (Jadeja & Pandya, 2014) • Modeling online reviews (Titov & McDonald, 2008)
Software Source Code Analysis	• Analyzing software evolution (Linstead et al., 2008) • Source code mining (Linstead et al., 2007a) • Mining eclipse developer contributions using author topic models (Linstead et al., 2007b) • Automating software traceability (Asuncion et al., 2010)

Outside Text Mining	• Scene analysis and abnormality detection (Varadarajan & Odobez, 2009) • Speech recognition (Chaney & Blei, 2012) • Image analysis (Fei-Fei & Perona, 2005; Sivic et al., 2005) • Biological data analysis (Pritchard et al., 2000) • Survey Data analysis (Erosheva, 2002)

Table 2 - Topic Modeling Studies

3.4.4 Software Implementations for Topic Modeling

There are several open source software implementations for different Topic Modeling algorithms. Table 3 provides a list of existing packages and their most salient features.

Author	**Package**	**Features**
McCallum (2002)	Mallet	• Built in Java • A comprehensive software package for various natural language process such as document classification, clustering, and information extraction • Implements LDA using Gibbs Sampling • Well supported and maintained. A very popular tool for topic modeling
Blei (2003)	lda-c	• Built in C • Extremely fast • Implements LDA with Variational Sampling
Steyvers (2005)	Matlab Topic Modeling Toolbox	• Built in Matlab • Well suited for research purposes but cannot be used for commercial applications as it is built in Matlab. • Implements LDA with Gibbs Sampling
Ramage (2009)	Stanford Topic Modeling Toolbox	• Built in Scala • Implements LDA • Allows visualization of topics in Excel • Not supported anymore by authors
Rehurik & Sojka	Gensim	• Built in python • Implements LSA, PLSA, and LDA using Gibbs

(2010)		Sampling • Well suited for commercial applications • Allows real-time online update of the topic model by feeding in more documents • Very compatible with other python natural language processing and machine learning libraries such as NLTK, NumPy, and Sciketlearn
Hornik & Grun (2011)	R package topicmodels	• Built in R • An R wrapper around lda-c by Blei (2003) • Very well suited for research • Works well with other R packages for NLP such as tm and textmineR • Well supported and very popular in the research community
Chang (2015)	R package lda	• Built in R • Implements LDA using Collapsed Gibbs Sampling • Implements other LDA topic models such as Supervised LDA, Correlated LDA, and Relational LDA • Works well with other R packages for NLP such as tm and textmineR

Table 3 - Topic Modeling implementations

3.5 Theoretical framework

The API service system can be broken down into API providers, API consumers, and end-users. API providers create APIs and publish them in the API service system. API consumers use those published APIs to create applications for the end-users. Value creation in the API service system flows from the provider to the consumer to the end-user. The link between the API service provider and the API service consumer is what this research focuses on.

The first step service providers take to create value for service consumers is publish their APIs in the API service system along with an API description. API descriptions are

composed of what the API does, or the service deliverable, as well as how it is beneficial to the consumer of the API. By communicating service deliverables and benefits, the API provider is communicating a customer value proposition to the API consumer.

The focus of this study is to use Topic Modeling to analyze API descriptions in the API service system of programmableweb.com. This allows the extraction of customer value propositions as communications of service deliverables and benefits.

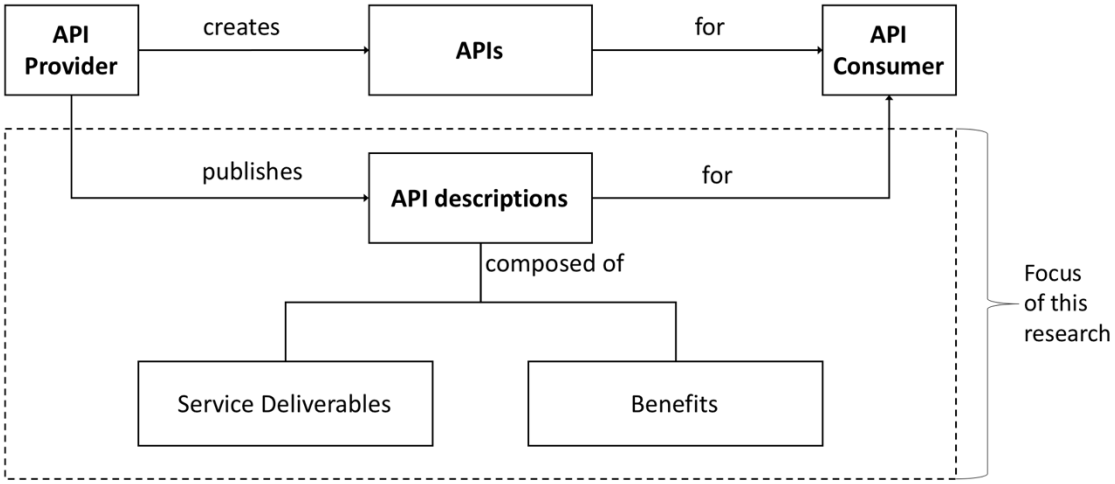

Figure 3 - Theoretical Framework

Figure 3 shows an illustrative diagram of the initial theoretical framework this study uses to guide data gathering and analysis. This framework is based on the literature review in Chapter 2.

3.6 Research steps

This section details the research steps taken to get the results and deliverables of this research.

3.6.1 Identify constructs

Identify and define the constructs and dimensions that are most salient to answering the research questions by: conducting a literature review, using observations from www.programmableweb.com and other web service systems such as Mashape[3], and using the researcher's own experience delivering an API program to a startup company in Ottawa. Section 2.4 covered this step.

3.6.2 Create initial theoretical framework

Organize the constructs in a framework that captures the initial relationship between the constructs. The framework will be used to focus efforts and guide the data collection and analysis. Section 3.5 covered this step.

3.6.3 Collect data

Using a Python script specifically designed for the purpose, crawl API descriptions, API names, tags, and submission dates from www.programmableweb.com.

3.6.4 Clean and import data into Mallet

Using a Python script, extract API descriptions from the database and create a text file

[3] https://market.mashape.com/

for each API where the file name is the API name and the text file contains the API description stripped from trailing and leading white space.

Using Mallet's import-doc command, import the files into mallet with the options of removing stop words and keeping the sequence of the documents.

3.6.5 Train the Topic Model

Run topic modeling on API descriptions in mallet with hyper parameter optimization and start from five topics with increments of 5. For each model create a table that includes the topic id and top 10 topic words. To find the model with the optimal number of topics:

1- Find the delta between the general topic interpretation between the current model and the previous model
2- Note the number of topics that include more than one theme
3- Note the number of topics that are semantically duplicates of other topics
4- The model with the optimal number of topics is the one that has the minimal delta with the previous topic, the minimal number of duplicate topics and the maximum number of topics with one theme.

3.6.6 Interpret the model

Prepare the model with the optimal number of topics for interpretation by removing topics that are too general to extract the necessary constructs using the metrics: tokens, document entropy, coherence, and rank 1 doc.

For the remaining topics create a table with

- Topic ID: This is the topic id assigned to the topic by Mallet they are digits that start from 0 for the first topic and go up to the number of topics. For clarity, increase all topic numbers by 1 so that topic number go from 1 to 40.
- Topic Category: This is a brief description that can uniquely identify the topics from each other based. This category is most similar to categories on programmableweb.com
- Top 10 Words: These are the top 10 words according to their distribution over the topic and are the output of Mallet.
- Service Deliverables: These are what the providers choose to offer to their API consumers. They are identified by looking at the verbs in the list of top 10 topic words and choosing the most meaningful verb that denotes what is actually delivered to the consumer.
- Benefits: These are what the APIs will benefit the API consumer.

Interpret the topic model by analyzing the semantic meaning of the top 10 words along with example API descriptions that are best represented by the topic.

3.6.7 Categorize customer value propositions

Create a table with the columns Topic ID, Topic Category, Service Deliverables, Benefits, and Customer Value Proposition Category.

The Service Deliverables and Benefits are under a general column of Customer Value Propositions.

The Customer Value Proposition Category is assigned based on the researcher's interpretation and analysis of the extracted information from the topic model.

3.6.8 Produce topic trend graphs

Time is an important dimension in studying processual and evolving phenomena such as networks (Halinen et al., 2012). Although the aim of this research is to identify and categorize customer value propositions in the API service system, the API service system is a continuously evolving network. The analysis of customer value propositions should include their evolution over time.

For each topic produce a graph that shows how the accumulative proportion of the topic presentation as a function of time. The topic proportion is calculated by identifying all documents that have this topic as the topic with the highest distribution to date divided by the total number of documents to date. Equation 1 gives document proportion value as a function of time.

$$TP_i = \frac{\sum_t d_{it}}{\sum_t d_t}$$

Equation 1 - Topic Proportion

where:

TP$_i$ → is the Topic proportion for topic ***i***

t → current time

d$_{it}$ → number of documents with topic ***i*** as the topic with highest distribution at time ***t***

d_t → total number of documents that have been posted until time ***t***.

Topic trend graphs were plotted for the period from June 2005 to July 2016. However, due to having very few APIs in the period from June 2005 to December 2005, the resulting graphs were uninterpretable as some of the curves started at 1.0 on the Y-axis while the rest of the curves varied between 0 and 0.7. To solve this problem, the graphs were looked at during the period from January 2006, to July 2016.

3.6.9 Categorization and statements of association

Create the final categorization model that categorizes the Customer Value Propositions based on the constructs of service deliverables and benefits.

Produce propositions and associations based on extant literature and findings.

4 Results

This chapter details the results of this research. The chapter starts by showing how the model with the optimal number of topics was chosen. It then describes the process used to prepare the model for interpretation. Next, the analysis and interpretation of the model are presented. Finally, topic trend graphs are produced and analyzed. The chapter ends with a brief summary of the findings. Where applicable, summaries of the results are presented and detailed results are referenced in the Appendices.

4.1 Identifying optimal model

The ultimate purpose of this part of the research is to extract the assets and service deliverables from a topic model that best represents the corpus of API descriptions in hand. While there are quantitative methods that determine the optimal number of topics, quantitatively significant models are not necessarily the most interpretable (Chang et al., 2009). Since the purpose is to find the model with the best interpretation, the exercise was qualitative and subjectively based on the researcher's ability to interpret the model. The researcher relied on three metrics to find the model with the optimal number of topics: the delta of new topics, the number of duplicate topics, and the number of topics with more than one theme. The delta between models in terms of the number of new topics added indicates saturation and that further models with higher number of topics will not add any new information. The number of semantically duplicate topics, that is two topics that are essentially the same theme, gives a further indication of saturation. An increased number of duplicates indicates reaching

saturation. Finally, the number of topics with more than one theme indicates the granularity of the model. More topics with more than one theme indicates that further models with higher number of topics will break certain topics into two or more topics. The model that minimizes all three is the best model.

Models were trained with increasing the number of topics from 5 to 45. Table 4 shows a summary of the results for each model. Appendix A.1 shows the detailed analysis for each of the models. In every iteration, topics were interpreted and given a 1 to 3-word phrase with the purpose of uniquely identifying this topic within the model and across models. Topics that included more than one theme were given two phrases separated by a comma. Each topic was given a letter denoting whether it is: a semantically **N**ew topic that was not in the previous model (N) or a semantic **D**uplicate of one of the topics in the current model (D). The number of topics that included more than one theme was also noted.

Iteration	Number of Topics	Delta New Topics	Number of Duplicates	Number topics with >1 theme
1	5	5	0	5
2	10	5	0	3
3	15	6	0	4
4	20	4	0	1
5	25	6	0	4
6	30	3	0	2
7	35	2	3	1
8	40	1	2	1
9	45	0	6	0

Table 4 - Results for optimal topic model

The delta of new topics was between 4 and 5 until model 6 with 30 topics and then

dropped significantly at models with 35 and 40, reaching 0 at 45 topics. The number of duplicate topics stayed at 0 until 30 topics and increased at 35 and 40. However, the number of duplicates spiked at 45. The number of topics with >1 theme stayed became 0 at 45 topics and was 1 for the 35 and 40 topic models. This makes the two models with 35 and 40 topics the best candidates for the best model. Since categorization is the ultimate goal of the research, more topics will improve the results of the categorization. However, a model with 100 topics will make categorization difficult since there will be many topics that are different than each other according to the LDA algorithm, but are the same in terms of human interpretation. Hence, model 8 with 40 topics was chosen to be the optimal topic for the purposes of this research.

4.2 Preparing model for interpretation

The Mallet toolkit produces some diagnostic metrics that helps with the interpretation of the model. This section provides the results from these metrics and explains how they were used to exclude some of the topics in the model before interpretation. The metrics were visualized using a visualization tool[4] obtained from the mallet website. Appendix A.2 includes a table with the diagnostics values.

4.2.1 Topic-Tokens metric

This metric measures the number of word tokens currently assigned to the topic. A number that is too high or too low relative to other topics signals that the topic may be

[4] http://mallet.cs.umass.edu/diagnostics.html

a bad one. If the measure is too small, it means that there are not enough observations to get a good sense of the topic's word distribution. A number that is too large means that this topic is too frequent, and may indicate that the topic words may be close enough to stop words.

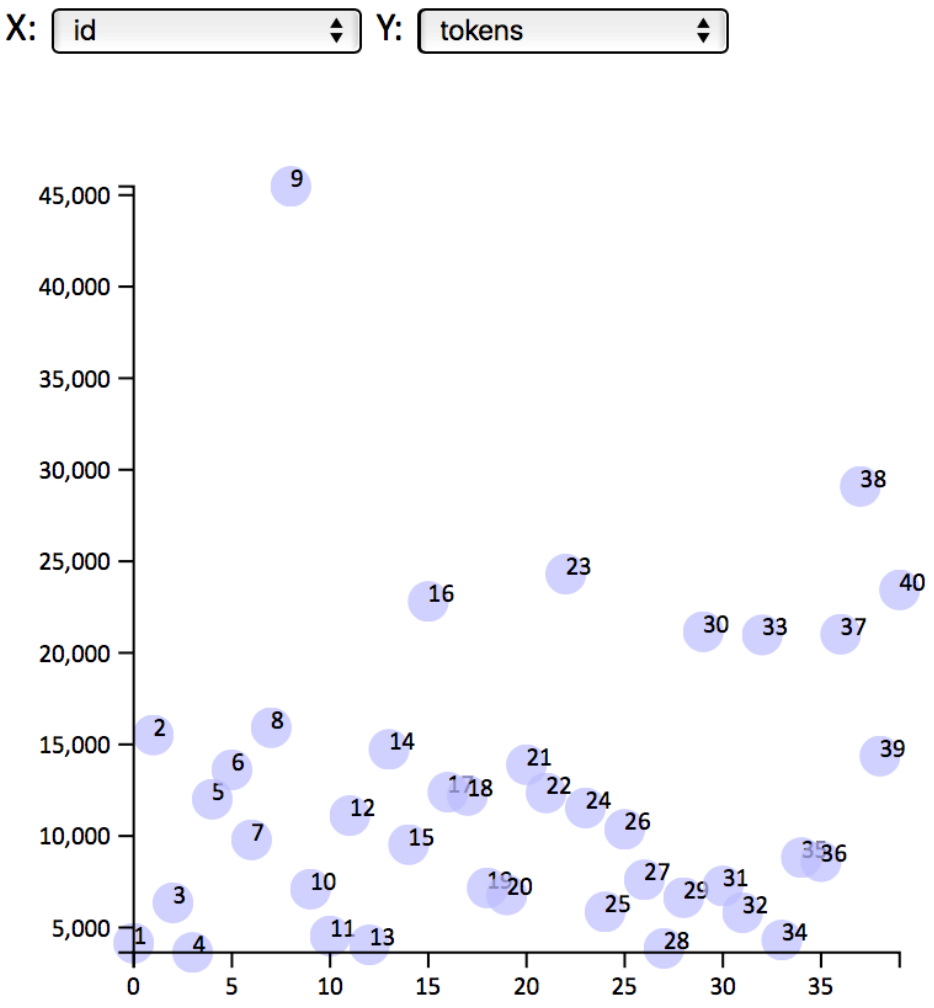

Figure 4 - Topic-Tokens metric

Figure 4 shows a scatter plot of topic Ids vs tokens. Relative to the majority of topics, topic 9 is at an extreme high with 45,475 tokens. This signals that the topic is too

frequent. The lowest token count is topic 4 with 3,637 tokens. Beyond topic 9, there are no major hikes or dips in the token metric. Topic 9 is:

{services applications web platform service integrate content offers access application}

The theme of the topic describes the general topic of web services, where words such as platform, access, content, and integrate are words that are too commonly used in any API description. According to the topic-tokens metric, topic 9 will be excluded from the final analysis of constructs.

4.2.2 Document Entropy metric

The document entropy metric measures the entropy of the probability of a document given a topic. Topics with lower entropy are concentrated in fewer documents, while higher entropy indicates the topic is evenly spread over many documents.

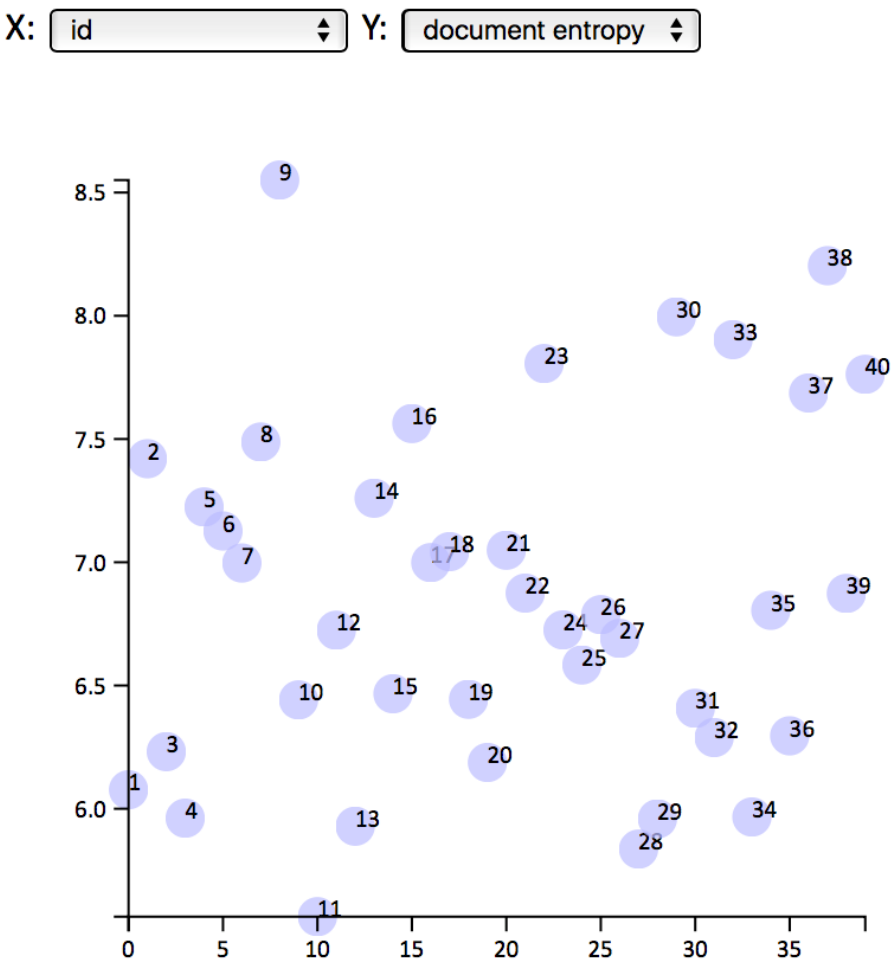

Figure 5 - Topic - Document Entropy metric

Entropy for model 8 ranged from 5.5621 to 8.5506. At the bottom of the scale is topic 11, and the topic with the highest entropy is topic 9. There were no major hikes or dips in the entropy metric. Figure 5 shows a scatter plot of topic ids vs entropy. Table 5 shows topics at the lower end of the scale and Table 6 show topics with the highest entropy.

Topic	Top 10 words	Interpretation
4	github software development application applications providers range wide integrate repository	Integration APIs for software code repositories
29	data sequence database protein sequences analysis biological soap research genes	APIs for accessing databases of protein sequence analysis
13	health medical information data healthcare care insurance providers fitness drug	APIs for accessing health information
28	print printing questions answer answers service design webknox knowledge models	APIs for question and answer sites
11	data ondemand barchart vehicle analysis number xml car request index	APIs for accessing vehicle information

Table 5 - Topics with low entropy

Topic	Top 10 words	Interpretation
16	methods service support functions submission including retrieval specific information status	Discusses the general concept of calling functions and methods as well as information retrieval as an integral part of web services
37	search data access database information results metadata library query research	Discusses the general concept of searching and database access
40	applications access functionality integrate include methods information retrieving managing create	Discusses the general concept of integrating functionality.
23	data management business software platform systems businesses services customer service	This topic discusses the general concept of web services from the business perspective
33	web site website url service page javascript websites link links	This topic discusses the general concept of web services from

		a more technical perspective.
30	responses json xml formatted restful calls data service protocol web	This topic discusses the general concept of web services protocols and specifically the RESTFul APIs
38	json key http rest authentication requests service access data account	This topic discusses the general concept of keys and authentication in web services. Showing how APIs in general are used
9	services applications web platform service integrate content offers access application	This topic discusses the general concept of integrating applications with platforms as the general purpose of web services.

Table 6 - Topics with highest entropy

Topics towards the higher end of the scale discuss general topics in web services such as their business aspect, their technical aspects, their protocols, how the web services are called, and what is the main purpose of calling the API. While these topics are interesting in that they give a nice break down of how APIs are presented on programmableweb.com, they are not helpful when it comes to trying to extract concepts such as assets exposed and service deliverables. Therefore, according to the entropy metric, topics 8, 16, 37, 40, 23, 33, 30, 38, and 9 will not be included in the list of topics for extracting assets exposed and service deliverable. However, they will be used as secondary points of data to triangulate findings.

4.2.3 Coherence metric

Coherence is a metric that measures the co-occurrence of the words in a given topic. The scores are log probabilities and therefore they are negative. Topics with scores closer to zero indicated that their words tend to co-occur together and therefore are more coherent. Figure 6 shows a scatter plot of topic-id vs coherence.

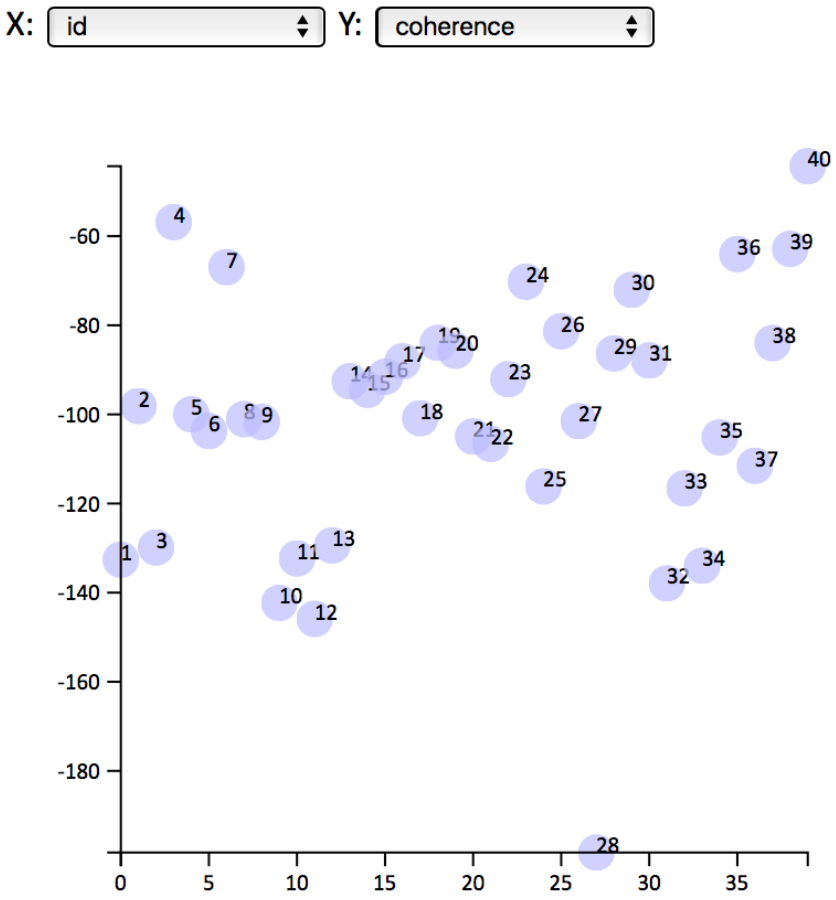

Figure 6 - Topic id vs coherence

Topic 28 shows significant incoherence relative to other topics in the corpus. According

to the coherence metric, topic 28 will be excluded from the final list of topics where assets exposed and service deliverables will be extracted.

4.2.4 Rank 1 Documents metric

Rank 1 documents is a metric that measures the amount of burstiness of a topic. A lower number indicates that the topic is a “background” topic that may arise from a certain style of writing or a certain context that is prevalent across the corpus. Figure 7 shows a scatter plot of topic-ids vs Rank-1-Documents.

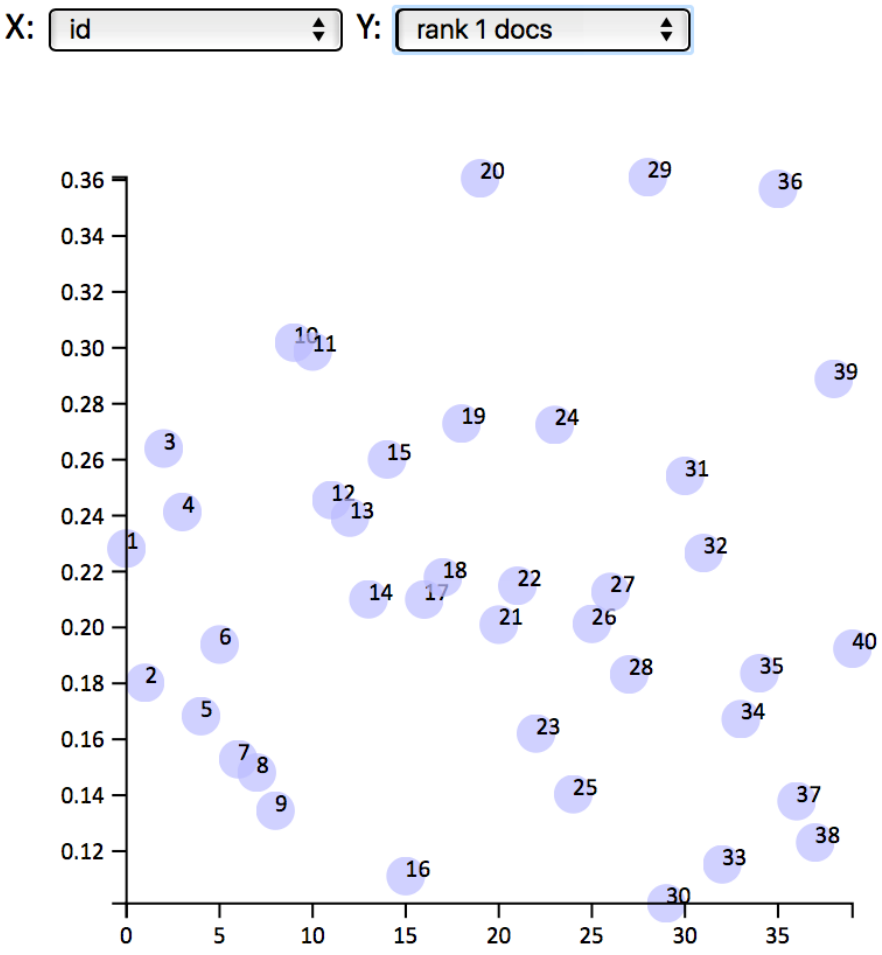

Figure 7 - Topic id vs Rank 1 docs metric

This metric confirms that topics 8, 9, 16, 23, 30, 33, 37, and 38 are generic topics that do not contain specific information about certain API types. Two more topics have low rank 1 doc measures and can be added to the list of background topics, they are topics 7 and 25. Table 7 shows topics 7 and 25

Topic	Top 10 words	Interpretation
7	website information soap calls provided format programmatically service xml documentation	Discusses the general concept of web services protocols and in this case SOAP.
25	code php java python javascript ruby codes net sample languages	A general collection of programming languages.

Table 7 - Topics excluded due to rank 1 docs metric

4.2.5 Topics to exclude from the final model

Table 8 shows each metric and the concluded topics to be removed as well as the associated reason.

Metric	Topics to be excluded	Reason
Tokens	9	Extremely high token count indicating the topic is too generic
Document Entropy	8, 16, 37, 40, 23, 33, 30, 38, 9	Topics with high entropy showing that they describe general aspects of web services.
Coherence	28	Extremely low coherence.
Rank 1 Documents	8, 9, 16, 23, 30, 33, 37, 38, 7, 25	Topics with low rank 1 documents indicating they

		are too general.
Unable to interpret	32	Example documents were inconsistent and not related.

Table 8 - Topic excluded from the final model

Excluding these topics from the final model leaves 26 topics to be analyzed for extracting customer value propositions. Table 9 shows the topics in the final model.

Topic	Words
1	property real affiliate estate search listings properties rental website program
2	social content twitter news media share facebook access data information
3	domain shipping delivery service services domains tracking mail dns customers
4	github software development application applications providers range wide integrate repository
5	mobile devices app apps device applications data android platform application
6	cloud data storage service server access servers monitoring amazon management
10	music events event information movie movies shows radio database artists
11	data ondemand barchart vehicle analysis number xml car request index
12	data travel booking market stock financial access information flight hotel
13	health medical information data healthcare care insurance providers fitness drug
14	marketing email campaigns platform analytics customers social customer advertising media
15	data weather information energy access time conditions location national solar
17	location map maps information data code address locations service mapping

18	product products information online shopping deals search access price store
19	voice call calls phone service applications services chat businesses integrate
20	game games data information sports players live player statistics scores
21	data information open access government state public u.s tax states
22	text language analysis content translation semantic sentiment machine words word
24	payment payments online card credit processing transactions account transaction cards
26	address data email addresses number service validation numbers phone verification
27	video videos content photos media photo audio share sharing upload
29	data sequence database protein sequences analysis biological soap research genes
31	data information traffic transit routes times public service bus time
34	project rest bot bots platform tools life projects goals management
35	images image files file pdf documents upload recognition document conversion
36	bitcoin exchange trading service account trade currency calls orders information
39	sms messages send messaging text mobile message bulk sending service

Table 9 - Topics in the final model

4.3 Model interpretation

A topic is a reduced set of dimensions that represent a prevalent theme in a corpus of text. The top 10 words in any given topic only reveals the general semantic meaning of the topic. Interpreting each topic could be done by the help of documents that are best represented by the given topic. To extract the information required for the required

analysis in this research, the use of actual API description documents had to be used in addition to the topic model. To narrow down a set of documents best represented by the given topic, a script was used to find all documents for every topic where the distribution for the highest topic is greater than 0.8 and the distribution for the second highest is less than 0.2. The script ran on the output-doc-topics file from Mallet that contains the documents and their topic distributions. The file was set to only include the distributions of the top 3 topics for each document. Appendix A.3 shows a sample of the output-doc-topics file.

This gives us documents that are mostly made up of one topic, which is the topic under examination. Using this criteria, topics 1 and 13 had no documents with more than 0.8 distribution. To get sample documents for these two topics, another script was written to find all documents in the corpus with topics 1 and 13 as their top most topics. The findings were sorted and the 3 documents with the highest topic distributions were selected as samples. Appendix A.4 shows the resulting examples along with topic distributions.

Topics were then interpreted by iteratively examining the top 10 words and samples of documents or API descriptions that are best represented by the topic. For each topic, the customer value proposition was extracted by interpreting the topic and the sample documents for the benefits and service deliverables that a consumer of this API would gain. The topics and the customer value propositions were then categorized. Table 10 gives a summary of the findings. Detailed analysis for each topic can be found in Appendix A.5.

Topic ID	Topic Category	Words	Customer Value Propositions		Customer Value Proposition Category
			Service Deliverables	Benefits	
4	Repository Management	github software development application applications providers range wide integrate repository	Provide access to software repository management functionality	Integrate software repository management functionality without building the infrastructure in house. It also allows developers to have programmatic access to the provider's services.	Enhance capability
6	Cloud Management	cloud data storage service server access servers monitoring amazon management	Access and management of cloud storage and Server management functionality	Utilize cloud capabilities and resources without acquiring infrastructure	Enhance capability
14	Digital Marketing	marketing email campaigns platform analytics customers social customer advertising media	Access to marketing and digital services capabilities provided by the API developer	Gives developers access to functionality that is costly to develop in terms of infrastructure and know-how	Enhance capability
17	Location and Mapping	location map maps information data code address locations	Access to locating and mapping functionality	Gives developers access to functionality that is hard to develop in	Enhance capability

		service mapping		terms of know-how	
19	Voice and Chat	voice call calls phone service applications services chat businesses integrate	Access to voice calling functionality	Ability to integrate functionality that is hard to develop because of the lack of know-how or the time needed for development	Enhance capability
22	Natural Language Processing	text language analysis content translation semantic sentiment machine words word	Gives access to a range of Natural Language Processing functionality such as keyword extraction, predict language sentiment, and extract relevant topics from text based on relevant categories.	The ability to do complicated NLP tasks without the need to develop the algorithms or invest in the infrastructure themselves.	Enhance capability
26	Verification	address data email addresses number service validation numbers phone verification	Validate email addresses, validate routing numbers, and validate postcodes.	Access to functionality that is hard to implement due to the high cost of gathering the necessary data and the lack of know-how	Enhance capability
27	Content	video videos content	Sharing and uploading	The ability to integrate	Enhance capability

		photos media photo audio share sharing upload	video, audio, and media content	functionality that is hard to develop into their applications	
29	Bio-informatics	data sequence database protein sequences analysis biological soap research genes	Provide a programmatic interface to functionality such as protein sequencing, protein sequence analysis, alignment of mass spectrometry images for comparison, and DNA sequencing.	Access to very complex machine learning algorithms that are hard to develop due to lack of know-how and the high cost of needed infrastructure.	Enhance capability
35	Format Conversion	images image files file pdf documents upload recognition document conversion	Allows developers to perform file format conversions between multiple file formats	Access to hard to develop functionality due to lack of know-how and time required	Enhance capability
36	Crypto-currency	bitcoin exchange trading service account trade currency calls orders information	Allows developers to perform cryptocurrency related tasks such as mining bitcoin, query exchange rate information and market statistics, and perform trading	Access to hard to gather data and hard to develop functionality	Enhance capability

			transactions like purchasing, holding, and selling bitcoins		
18	E-Commerce	product products information online shopping deals search access price store	Searching Product pricing and deals	Allows developers to become affiliates and resellers to the main service provider.	Enhance capability
3	Logistics	domain shipping delivery service services domains tracking mail dns customers	Integrating shipping, tracking, and delivery functionality	The ability to provide shipping and delivery services without owning a shipping company or building a shipping infrastructure	Enhance capability
10	Media	music events event information movie movies shows radio database artists	Ability to programmatically search databases of music, movies, shows, artists, and events	Ability to use information in applications that they could not have access to previously	Increase information resources
12	Travel and Finance	data travel booking market stock financial access information flight hotel	Booking Flights and Hotels and access to flight and hotel information.- Access to financial	Access to financial and travel data gathered, aggregated, and analyzed by the provider	Increase information resources

			information and performing financial services tasks such as currency conversion and analysis		
15	Weather	data weather information energy access time conditions location national solar	Access to weather related information	Gives developers access to data that requires large infrastructure and knowledge investments to gather, aggregate, and store.	Increase information resources
20	Sports Statistics	game games data information sports players live player statistics scores	Access to Sports, players, and game information such as statistics and metrics	Getting access to hard to collect and aggregate sports data	Increase information resources
21	Open Government	data information open access government state public u.s tax states	Access to government open data. Provide functionality for modeling and searching open data.	Allow data providers to model and open their data in a standardized and consistent way and allow developers to access hard to collect and aggregate data	Increase information resources

31	Transit Data	data information traffic transit routes times public service bus time	Access to real-time bus, subway, and train information such as location and schedule	Access to hard to gather information that only a city can provide.	Increase information resources
13	Medical Information	health medical information data healthcare care insurance providers fitness drug	Access to medical information from multiple sources	Gives developers the ability to access multiple resources aggregated and connected to by the provider.	Increase Information Resources
1	Real Estate	property real affiliate estate search listings properties rental website program	Searching for real estate information	Access to expensive to collect data they previously did not have access to	Increase information resources
5	Mobile Integration	mobile devices app apps device applications data android platform application	Integrate mobile applications with functionality provided by a platform	Allows developers to customize and personalize the use of the product.	Person-alization
34	Workplace Collab-oration	project rest bot bots platform tools life projects goals management	Enable the users of the products to customize and integrate project management, collaboration and chatting tools provide by the	The main benefit these APIs provide is interoperabilit y and personalizatio n of the main service provided by the vendors.	Inter-operability

			vendors with other applications		
2	Social Media	social content twitter news media share facebook access data information	Sharing content on multiple social media sites	Reduces the cost of implementation as they only have to integrate with one supplier and they get access to multiple social media sites at once	Linking Services
24	Payment	payment payments online card credit processing transactions account transaction cards	Provide payment processing functionality for digital and real currencies	The ability to process payments without having to deal with banking systems, compliance, fraud, and security	Linking Services
39	SMS Messaging	sms messages send messaging text mobile message bulk sending service	Access to SMS messaging functionality and connectivity to multiple mobile network operators around the world	Ability to connect to multiple mobile network operators at once, reducing the cost of connectivity and integration	Linking Services
11	Vehicle Registration	data ondemand barchart vehicle analysis	Access to Vehicle registration information from multiple	Access to data they could not get access to before as well as querying	Linking Data

		number xml car request index	countries	multiple sources of data at once.	

Table 10 - Model interpretation and categorization summary

4.4 Topic trend graphs

Topic trend graphs were produced by plotting the proportion of the cumulative number of documents mostly composed of a given topic versus the overall cumulative number of documents in a given date in time. Two types of graphs were produced: individual graphs and aggregate graphs. Individual graphs were produced by plotting a graph of topic proportion versus time for each topic for the given topic ids, while aggregate graphs were produced by plotting the sum of topic proportions for a given set of topic ids versus time. Individual graphs are useful in identifying trends among topics that belong to a certain customer value proposition category, and aggregate graphs are useful in identifying trends among categories of customer value propositions.

4.4.1 Aggregate topic trend graphs for all categories

Figure 8 shows the aggregate plots of topic trend graphs for all the categories that emerged in section 4.3. They category Increasing Capabilities takes the highest share of the documents at around 27% in 2016. It increased steadily from 20-25% in 2006/2007. The second place is occupied by the category of Increasing Information Resources. This category showed a constant and steady variation around the 13% mark. Linking Services comes at third place with 9% in 2016. These three categories combined are taking around 50% of the overall document count. The next three categories are

Personalization at 2.5%, Linking data and Interoperability both at 1%. The next three categories combined form 5.5-6% of the document share, indicating that they fall more towards niche categories than mainstream categories. One thing to note about these curves is, except for Linking Data, the rest of the curves are mostly steady throughout time. However, as will be shown in the next sections, within these categories there are many variations at the individual topic levels. This indicates movement within the industry from niche to mainstream, and vice versa.

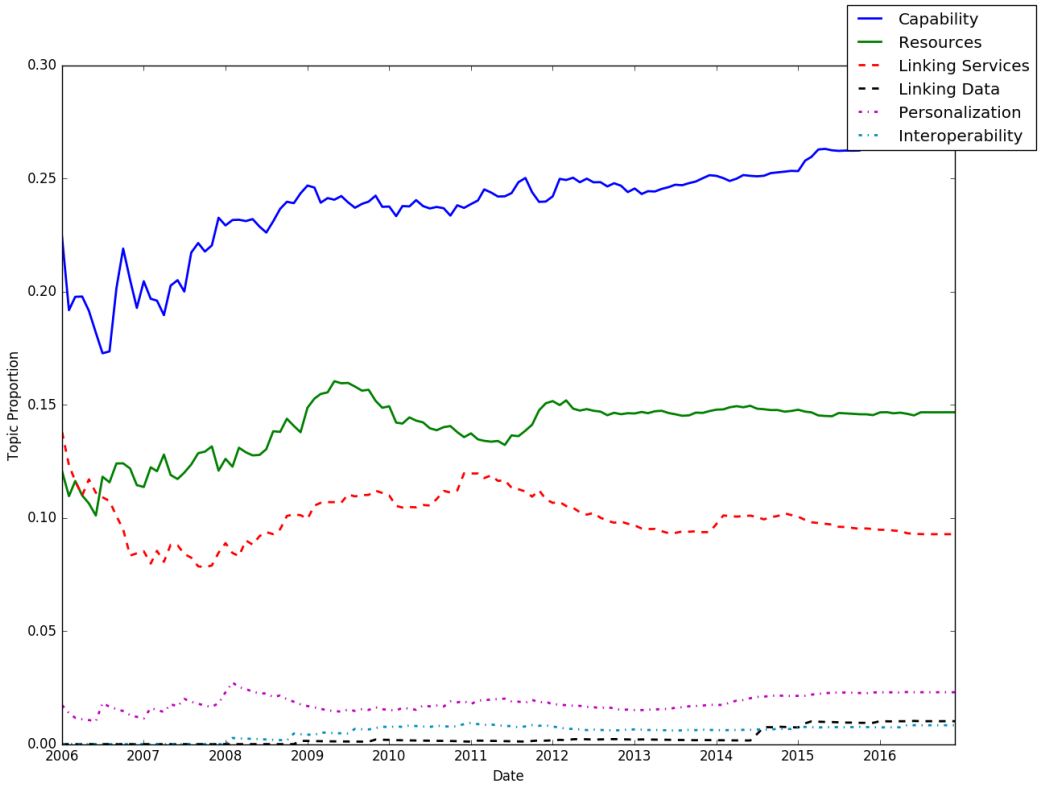

Figure 8 - Aggregate topic trend graphs for all categories

4.4.2 Individual topic trend graphs for category: Increasing Capability

Figure 9 shows plots for individual topic trend graphs in the category Increasing

Capability showing a lot of variation during the period 2008 to 2009. This is expected as the API service system was just forming and getting in shape. A period of stability followed between 2009 to 2011, where most categories were well established and in position. During this period, Location and Mapping had the highest share of the topics followed by E-commerce. This result is again expected as mapping and e-commerce APIs were the most common use cases in the API economy.

In mid-2011 and beginning 2012 a lot of movement occurred within the API service system where niche areas rose significantly taking their positions among the top categories in the service system. The categories of Bioinformatics and Cryptocurrency saw significant increase in the document share. Also, Digital Marketing, Cloud Management, and Natural Language Processing saw increases in this period. In 2015, the category of Repository Management saw a jump, however it remains a niche area in the API service system.

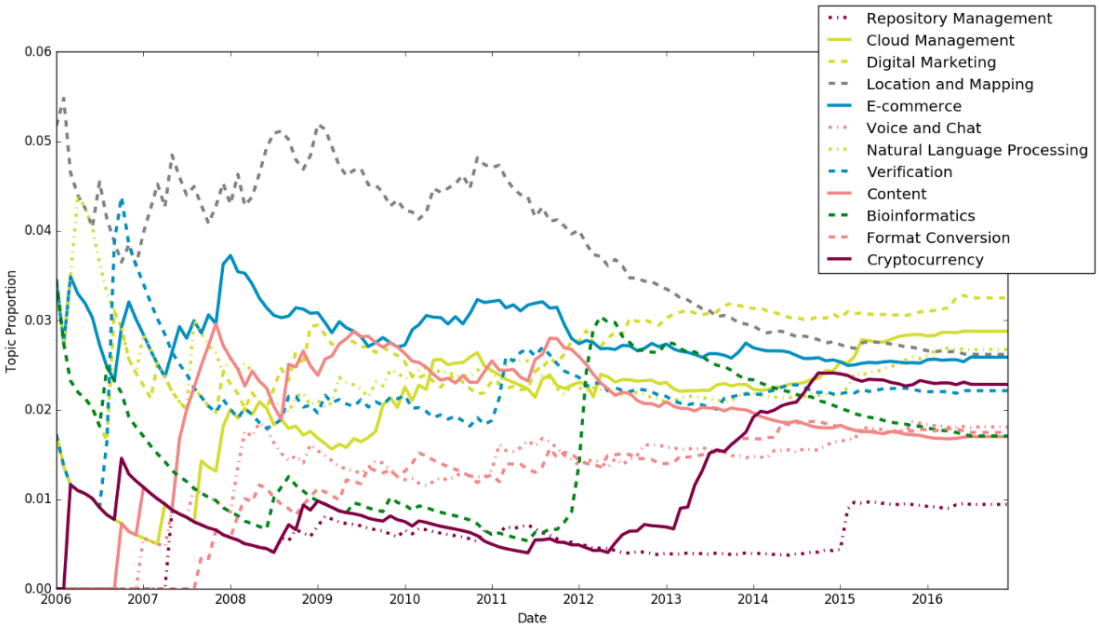

Figure 9 - Individual topic trends for Increasing Capability.

4.4.3 Individual topic trend graphs for category: Increasing Information Resources

Figure 10 shows topic trend graphs for individual topics in the category of Increasing Information Resources. Similar to Increasing Capability, this category saw a period of high movement between the inception of the API service system and early 2009. This was followed by a relatively stable period between 2009 and mid 2011. After mid 2011 some categories saw upward movement to join the mainstream.

This graph can be seen as composed of two clusters: 1) Travel and Finance, Open Government, and Media, 2) Weather, Real Estate, Transit Data, and Sports Statistics. During the stable period, cluster 1 took the highest share of the API service system, while cluster 2 came in second place. Medical Information was the lowest among all of them. Between mid-2011 and 2012 Weather and Transit Data from cluster 2 rose,

coming close to the declining Travel and Finance, and Media from cluster 1. In cluster 1 Open Government increased as well, which correlates well with Transit Data. In cluster 2, Real Estate declined, while Medical Information increased coming close to Real Estate.

These movements show that mainstream categories such as Travel and Finance, Media, and Real Estate declined compared to emerging categories such as Weather, Open Data, and Medical Information.

Figure 10 - Individual topic trend graphs for Increasing Information Resources

4.4.4 Individual topic trend graphs for category: Personalization and Interoperability

Figure 11 shows the plots for individual topics in the categories of Personalization and

Interoperability. The graph shows a steady increase in both categories since the beginning of the API service system. Mobile Integration takes the larger share of the service system. An interesting observation to note in this graph is the sharp peak in Mobile Integration around 2008. This was the period where the global financial crises occurred, this peak shows a sudden drop in Mobile Integration use cases from the exponential increase it was observing. However, Mobile Integration which is categorized as Personalization saw a comeback in 2009 increasing steadily since then.

Figure 11 - Individual topic trends for Personalization and Interoperability.

4.4.5 Individual topic trend graphs for categories: Linking services, and Linking data

Figure 12 shows individual topic graphs for categories Linking Services and Linking Data. The graph shows similar trends to the previous graphs, where the period between 2006

and 2009 show large variability, the period between 2009 and 2012 show steadiness and the period after 2012 show more movement. Social Media here shows the largest movement at the mid 2011 mark declining to join the remaining topics. Both SMS Messaging and Vehicle Registration saw a step increase around 2014.

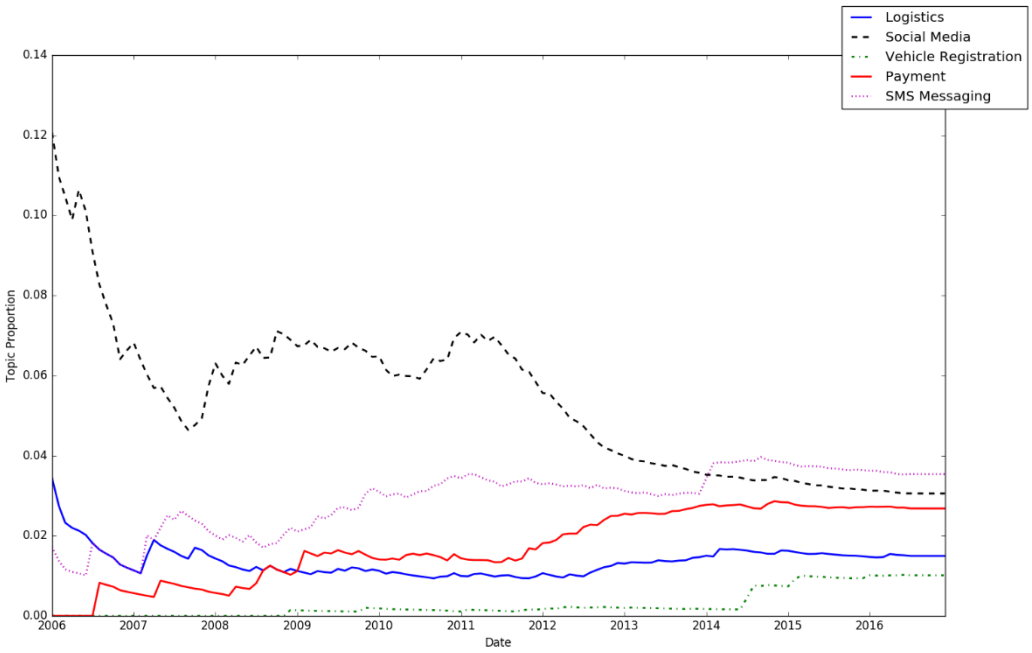

Figure 12 - Individual topic trend graphs for Linking Services and Linking Data

4.4.6 Individual topic trend graphs for all topics

Figure 13 shows individual topic trend graphs for all topics. In this graph the three periods reported in previous graphs are very obvious. The first period is between 2006 to 2009. This period was the inception of the API service system, and many topics got introduced making movements and variability high in this period. Between 2009 and mid 2011 was a period of relative steadiness in all topics. After 2012 more movement occurred and all topics converged coming very close to each other.

The period between 2009 and 2011 showed a distribution in topics that is characteristic of a power law distribution where few topics had the highest share and many topics had smaller shares. However, in 2016 this distribution seems to disappear showing a steady declining distribution between all topics.

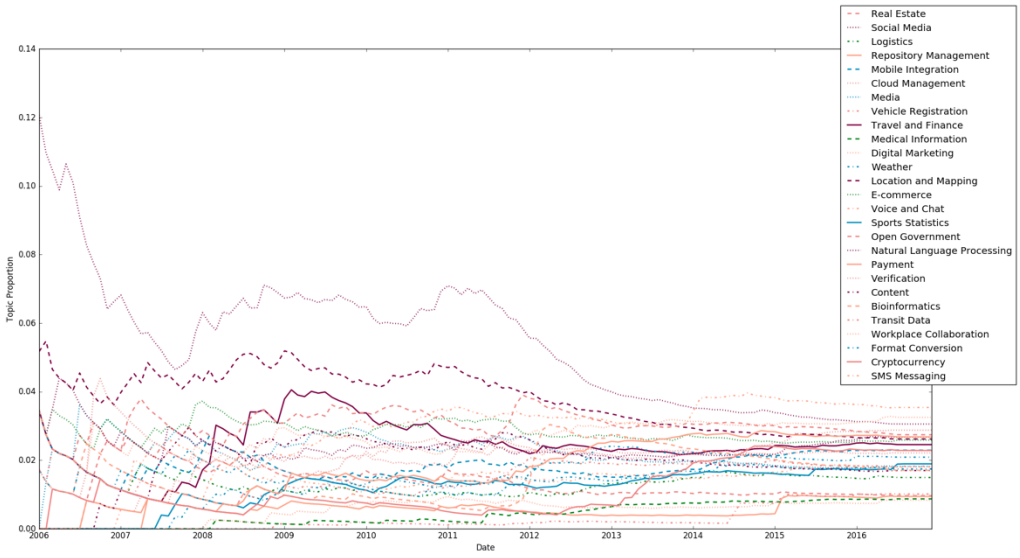

Figure 13 - Individual topic trend graphs for all topics.

4.4.7 Long tail graphs

The observation from section 4.4.6 of a disappearing power law like distribution signals a long tail in the API service system among topic distributions. This calls for plotting topic proportions versus topic ranks to see how the service system has evolved. The curves would need to be plotted at a snapshot of time and two snapshots were chosen: June 2009, and January 2016. June 2009 was the beginning of the steady period where most topics have taken their position in the service system and not much movement is going on. January 2016 is a later stage of the second high movement period where

topics movement has also settled.

Figure 14 shows the two curves plotted on the same graph. The June 2009 curve is typical of a long tail curve where few topics have the largest share of the API service system and the rest have lower shares. The January 2016 curve however shows a steady declining slope where topics have steady decreasing distributions. These results show that the API service system had a long tail distribution that has flattened out.

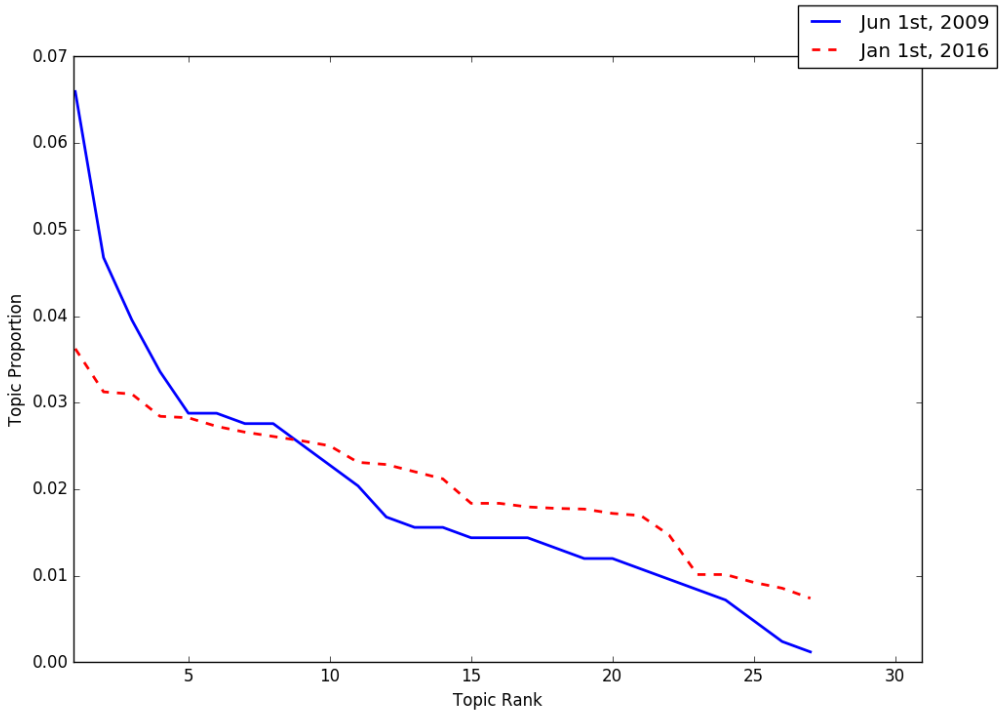

Figure 14 - Topic proportions vs Topic Rank

4.4.8 Long tail topics and topic proportions

To further break down what happened in the long tail a table with topic categories and topic proportions was created for each period snapshot. The topics were sorted decreasing order of topic proportions to reflect the long tail curve. To identify the head

and the tail of the long tail the area under the curve was calculated for each topic going from left to right (Area Increasing) and from right to left (Area Decreasing). The area was calculated by adding the topic proportions up to the current topic. The two topics around which the area increasing and the area decreasing cross each other was marked as the cutoff between the head and the tail. Essentially this marks the cutoff where the area under the curve for the head is equal to the area under the curve for the tail.

Table 11 shows the topic proportions for the June 2009 snapshot. The head included topics that were considered mainstream for the longest time. Topics such as Social Media, Mapping, Travel and Finance topped the head with the largest share of APIs. Closer to the head were Media and SMS Messaging both with topic proportions that are very close to topics in the head. Due to the way the head cut-off was calculated these topics did not make it into the head, however Media for example has the same distribution of documents as Digital marketing.

Jun-09	Topic ID	Topic Category	Topic Proportion	Area Increasing	Area Decreasing
Head Hits	2	Social Media	0.065947242	0.066	0.540
	17	Location and Mapping	0.04676259	0.113	0.474
	12	Travel and Finance	0.039568345	0.152	0.427
	21	Open Government	0.033573141	0.186	0.387
	27	Content	0.028776978	0.215	0.354
	18	E-commerce	0.028776978	0.243	0.325
	14	Digital Marketing	0.027577938	0.271	0.296
Tail Niches	10	Media	0.027577938	0.299	0.269
	39	SMS Messaging	0.025179856	0.324	0.241
	22	Natural Language Processing	0.022781775	0.347	0.216
	26	Verification	0.020383693	0.367	0.193
	6	Cloud Management	0.016786571	0.384	0.173

	1	Real Estate	0.01558753	0.399	0.156
	24	Payment	0.01558753	0.415	0.140
	5	Mobile Integration	0.014388489	0.429	0.125
	15	Weather	0.014388489	0.444	0.110
	20	Sports Statistics	0.014388489	0.458	0.096
	19	Voice and Chat	0.013189448	0.471	0.082
	35	Format Conversion	0.011990408	0.483	0.068
	31	Transit Data	0.011990408	0.495	0.056
	3	Logistics	0.010791367	0.506	0.044
	29	Bioinformatics	0.009592326	0.516	0.034
	36	Cryptocurrency	0.008393285	0.524	0.024
	4	Repository Management	0.007194245	0.531	0.016
	34	Workplace Collaboration	0.004796163	0.536	0.008
	13	Medical Information	0.002398082	0.538	0.004
	11	Vehicle Registration	0.001199041	0.540	0.001

Table 11 - Topic categories vs topic proportions in the long tail curve of June 2009

Table 12 shows the topic categories vs topic proportions for the period of January 2016. The head of the long tail curve in 2016 was expanded by SMS Messaging, Payment, and Natural Language Processing. In addition, Cryptocurrency, Mobile Integration, Verification, and Weather came very close to the head. We can also observe that topic distributions came closer to each other with the difference between the highest and the lowest topic distributions being 0.023 in 2016 from 0.065 in 2009.

Jan-06	Topic ID	Topic Category	Topic Proportion	Area Increasing	Area Decreasing
Head - Hits	39	SMS Messaging	0.036233691	0.036	0.559
	2	Social Media	0.031247403	0.067	0.523
	14	Digital Marketing	0.030998089	0.098	0.492
	21	Open Government	0.02842184	0.127	0.461
	6	Cloud Management	0.02825563	0.155	0.432
	24	Payment	0.027258373	0.182	0.404
	17	Location and Mapping	0.026593534	0.209	0.377

	22	Natural Language Processing	0.026094906	0.235	0.350
	18	E-commerce	0.025596277	0.261	0.324
	12	Travel and Finance	0.025014543	0.286	0.299
Tail - Niches	36	Cryptocurrency	0.023103133	0.309	0.274
	5	Mobile Integration	0.022853819	0.332	0.251
	26	Verification	0.022022771	0.354	0.228
	15	Weather	0.021191723	0.375	0.206
	19	Voice and Chat	0.01836616	0.393	0.184
	31	Transit Data	0.01836616	0.412	0.166
	29	Bioinformatics	0.017950636	0.430	0.148
	35	Format Conversion	0.017784426	0.447	0.130
	10	Media	0.017701321	0.465	0.112
	20	Sports Statistics	0.017202693	0.482	0.094
	27	Content	0.016953378	0.499	0.077
	3	Logistics	0.014709549	0.514	0.060
	11	Vehicle Registration	0.010138785	0.524	0.045
	1	Real Estate	0.010138785	0.534	0.035
	4	Repository Management	0.009224632	0.543	0.025
	13	Medical Information	0.008559794	0.552	0.016
	34	Workplace Collaboration	0.007396327	0.559	0.007

Table 12 - Topic categories vs topic proportions in the long tail curve of Jan 2016

4.5 Summary of findings

This research presents an exploratory study and methodology for studying the practical problem of articulating customer value propositions in the API service system. Topic Modeling was used as text mining and knowledge discovery technique to analyze a corpus of API descriptions obtained from programmableweb.com. Two types of analyses were run on the resulting Topic Model: topic interpretation and topic trend analysis.

Topic interpretation delivered a categorization of customer value propositions in the API service system. Six categories of customer value propositions were identified:

- **Increasing Capability** by allowing API consumers to do things they could not do before such as mining cryptocurrency, verifying email addresses, or sequencing proteins.
- **Increasing Information Resources** by allowing API consumers to access data resources they could not access before such as sports statistics, transit schedules, or weather data.
- **Linking Services** by connecting the API consumer with multiple services from one place, thus reducing the cost of connectivity.
- **Linking Data** by giving the API consumer access to multiple databases or sources of data, thus reducing the cost of aggregating data.
- **Allowing Interoperability** by allowing the API consumer to integrate existing applications and services with a product or a service provided by the API provider
- **Allowing Personalization** by allowing the API consumer to build custom applications that personalize a service or a product provided by the API provider.

Topic trend analysis delivered a long term distribution over topic proportions in the API service system. Analysis of the long tail curve revealed that curve is flattening and moving away from a long tail to becoming a declining slope. The flattening of the long tail is caused by niche areas becoming mainstream as well as convergence in the distribution of areas overall.

5 Discussion

This research focused on studying customer value propositions in the API service system using Topic Modeling as a data analysis technique. Using the interpretation of the Topic Model, customer value propositions were categorized into six categories. By analyzing Topic trends over time, a long tail distribution of topics was revealed in the API service system. This section discusses the findings by: (i) constructing a categorization model for customer value propositions in the API service system, (ii) taking a closer look at the flattening of the long tail and how it affects the current state of the API service system, and (iii) proposing managerial recommendations for three groups of stakeholders.

5.1 Categorization Model of Customer Value Propositions in the API Service system

Topic modeling enables the discovery of latent semantic themes in a corpus of text. This implies that the topics discovered are abstract themes revealing the hidden semantic structure of the text. Every API description in the data corpus is composed of some proportion of each of the topic instances listed in this categorization model. This categorization model enables the classification of an API description based on the category of customer value proposition to which it belongs.

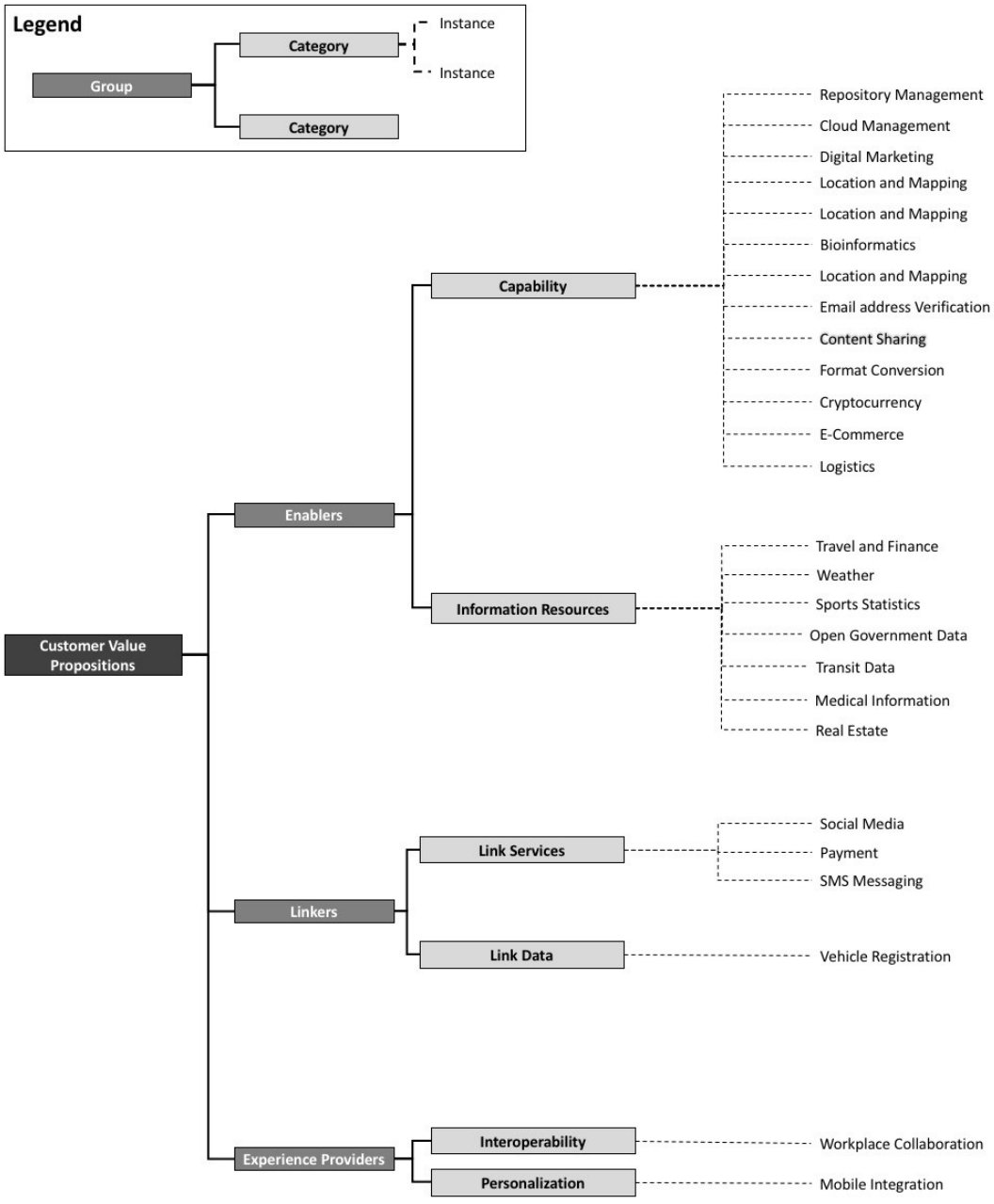

Figure 15 - Customer value propositions categorization model

Figure 15 shows a categorization model of customer value propositions in programmableweb.com's API service system. The model is composed of 3 groups of

customer value propositions: Enablers, Linkers, and Experience Providers. Under each group there are two categories, and for each category the instances of topics that make this category are listed.

5.1.1 Enablers

This group of customer value propositions promise to enable the customer to do something they are not able to currently do and for this reason they are called *enablers*. Enablers form the vast majority of customer value propositions in the API service system and are broken down into two categories: increasing capabilities, and increasing information resources.

Providers with value propositions that promise to increase capabilities provide services that are heavy on functionality such as cloud resource management, mining cryptocurrency, file format conversion, and taking care of shipping logistics. The APIs created by these providers enable the API consumer to use the functionality without the need for building it themselves, thus increasing the capability of the API consumer. The complexity of the functionality provided can vary tremendously. The complexity of the functionality can range from simple tasks such as uploading a file or a video to extremely complex tasks such as mining bitcoins and protein sequence analysis.

The second category in the enablers group is value propositions that promise to increase information resources. Providers communicating these value propositions create APIs that give access to some data or information resource such as weather data, open government data, or sports statistics. Information resources vary in the level of

insight they enable the API consumer to gain. At one end there is crude data resources such as open government data, or a list of hotels in a certain area. On the other end are information resources with high levels of insight such as financial market trends, and sports analyses. In addition to the level of insight, time is a major factor in the type of information resources provided. The information can be changing real-time such as in the case of hotel room prices, stock prices, or live game statistics and metrics. Or it can be on demand and query driven like hotel locations, real estate listings, sports profile information, and medical insurance information. It is important to note that even hotel locations and real estate listings change over time, but the rate of change and the method of update is different than things like stock prices and live game statistics. Real-time driven information resources are pushed to the API consumer, while on demand information is pulled by the API consumer.

5.1.2 Experience Providers

The second group of customer value propositions in the API service system allow the API consumer to personalize, or interoperate with a product or a Software as a Service. The API in this case is not the primary focus of the service provider, but is used to enhance the experience of a product or a service that drives the core value. API providers promise to enhance experience by either increasing the interoperability or personalization of the product or service they are providing to their customer. Interoperability refers to the ability to integrate existing applications and infrastructure with the new product. While personalization allows the customer to modify the use of the product or service in ways that suit their needs. Personalization can be achieved by

allowing customers to build their own mobile applications using the provided API.

5.1.3 Linkers

The third group of customer value propositions allow API consumers to connect to multiple services and multiple sources of data. API providers with this type of customer value propositions promise to link API consumers to multiple services such as multiple social media providers, or multiple sources of data such as vehicle registration database from different countries, through one interface. API providers promise to take the headache of dealing with things like different regulations, compliance, aggregation of data, and heterogeneous interfaces away from API consumers.

The two categories of customer value propositions that make up the linkers group are reducing the cost of connectivity, and reducing the cost of data aggregation. One example of reducing the cost of connectivity is value propositions by SMS messaging API providers. API providers that provide SMS messaging services establish connection with tens or hundreds of mobile network operators around the world and allow the API consumer to send a text message to any phone around the world from one interface. There is a high cost involved in establishing all these connections. This cost is mainly dealing with different regulations, establishing multiple agreements, interfacing with operators that implement different standards and making sure the messaging service is always up and running. SMS Messaging API consumers do not have to do all of these steps, and hence their cost of implementing SMS messaging functionality is reduced.

An example for linkers that promise to reduce the cost of aggregating multiple sources

of data is customer value propositions by vehicle registration API providers. Vehicle registration API providers gather, administer, and maintain large databases of vehicle and vehicle registration related information. Their value propositions promise API consumers to reduce the cost involved in accessing multiple databases at once.

5.2 The long tail of customer value propositions

Studying the API service system using network science techniques based on the links between mashups and APIs revealed a power-law distribution of mashups to APIs. Researchers linked this finding to the idea of a “long tail”, where a distribution is characterized by a large number of low frequency occurrences and a very small number of high frequency occurrences (Woodard, 2009).

Studying the API service system using the lens of Topic Modeling revealed another long tail distribution. This long tail is the distribution of topic proportions over topics in the API service system. It reflects how customer value propositions are communicated in the service system as well as how areas of the API service system have evolved. Earlier in the evolution of the programmableweb.com API service system, the distribution of topic proportions followed a long tail curve as reported in section 4.4. In 2009, the head of the long tail included areas such as Social Media, Mapping, Finance, and E-commerce. These were the four main areas where APIs strived during that period of time. In 2016, this demographic changed in two ways. First, areas such as SMS messaging, Cloud Management, and Natural Language Processing jumped in ranks and joined the head, causing the head to expand. Second, the distributions along the tail got closer to each

other causing the curve as a whole to flatten out and become more of a declining slope than a long tail.

The emergence of SMS Messaging, Cloud, and Natural Language Processing can be seen to correlate with the emergence of the Mobile, Cloud, and Big Data and Machine learning industries. This implies that the API industry is a secondary industry that reflects what occurs in other industries rather than influence it. Thus, primary innovations that create new industries are scarce and the API economy is not reaching its full potential.

The flattening of the long tail could be argued to be due to the saturation of APIs in every category in the API service system. When the API service system started some categories like Social and Mapping APIs enjoyed wide popularity in terms of number of APIs and number of uses. At this point, other categories such as Cryptocurrency and Bioinformatics emerged with the number of posted APIs large enough to cause the flattening of the long tail. This implies that the supply is increasing in the API service system.

The increase of supply in the API service system does not necessarily result in the shift of demand towards that supply. According to the long tail theory, “filters” are needed to shift the demand down the tail. Filters are people or software that help consumers find what they need in an increasingly saturated market (Anderson, 2006). The stagnation of mashup listings indicates that the demand is not being shifted towards the rising tail and that API service systems such as programmableweb.com are not filling the “filters” role

properly in the API economy.

5.3 Connecting supply and demand in the API economy

There are three forces that shape the long tail of an economy: (i) democratizing production, (ii) democratizing distribution, and (iii) connecting supply and demand (Anderson, 2006). Open APIs based on the RESTful architecture have democratized the production of web services by making it increasingly simple to create APIs. Directories such as programmableweb.com' API directory and Mashape's API marketplace have democratized distributions by aggregating and listing APIs. The missing link in the API economy are the filters that connect supply and demand.

To demonstrate how API service systems can play the role of filters effectively the we will take the angel of differentiation in the API service system. Differentiation is the realm of a properly communicated customer value proposition (Anderson et al., 2006). There are three models by which value propositions can be communicated: (i) all benefits, (ii) favorable points of difference, and (iii) resonating focus.

Best practices in customer value propositions entail presenting and communicating value propositions using the resonating focus model. In this model the API provider should communicate the one or two points of difference and a point of parity whose improvement will deliver the greatest value to the customer in the foreseeable future (Anderson et al., 2006). The main idea behind the resonating focus is to answer the customer question: "What is most worthwhile to keep in mind about your offering?". Resonating focus requires very high level of knowledge about one's own offering, the

next best alternative and the customer.

To enable API service providers to communicate resonating focus value propositions to their potential customers, APIs need to move away from being a white technology where the emphasis is on features and characteristics of the API, to a branded technology where the emphasis is on the brand behind the API. To make the move API service providers need to focus on their brands and API service systems need to establish the proper environment for API service providers to communicate their brands. This is one way the API service systems can play the role of a filter connecting supply and demand in the API economy.

5.4 API service systems from aggregators to filters

In a recent post, programmableweb.com announced a new API data model that reflects the current trends in the API service system. The new model reflects things like the rise of Software Development Kits(SDKs), and the proliferation of different types of APIs such as product embedded APIs and browser APIs (Berlind, 2016). However, the new described data model still focuses on listing aspects of the API that do not allow API providers to differentiate themselves. The new model consists of information such as the API name, the provider's home page, the API portal's home page, and the API type according to what programmableweb.com defines as API type. This information is only helpful in directing API consumers to the API's homepage where more information can be found about the API and the provider.

Another famous API service system is Mashape which goes a step further than pure

listings by enabling developers to test drive the APIs during the search process. This is done through an interface to the API inside Mashape's website where a developer can make test calls to the API to see if the results are what they are expecting. However, this approach is still short of enabling API service providers to reveal their brand and communicate resonating focused value propositions to customers.

Studies on web service discovery mechanisms can hint to some ideas that can enable the formation of an API service system that supports resonating focus value propositions. Researchers have suggested leveraging social information such as who used the API to enable the discovery of APIs (Torres et al., 2011), graph based methods for personalizing API selection (Dojchivnovski et al., 2012), and API recommendation based on network prediction (Huang & Tan, 2014). These studies show that knowing how the API ranks with respect to other APIs can be used to recommend APIs for potential developers. Therefore, an API service system that supports branding of the API based on who used it, what was made with it, famous success stories, and star like ranking is needed.

The current state of API service systems is similar to SourceForge and similar Open Source Software (OSS) directories in the early stages of OSS. At that time, basic "yellow pages" like directories of OSS and free software was all what is available in the market. Later, GitHub came along and took the OSS industry by a storm eventually taking over the OSS industry and becoming the most used OSS service system. GitHub provides OSS developers more than just listings. GitHub provides features such as starring, wikis, blogs, dedicated website, and more. All of these features allow OSS developers to create

communities around their OSS, making differentiation possible. This makes GitHub an service system of service systems rather than just a directory of OSS repositories. Hence, coming closer to playing the role of filters connecting supply and demand.

5.5 Managerial recommendations

Next, results of this study are used to provide some recommendations to three groups of practitioners: API service systems, API service providers, and entrepreneurs.

5.5.1 Recommendations for API service systems

API service systems are at a unique position to become filters in the API economy and meet a much needed need of connecting supply and demand. Current API service systems such as programmableweb and Mashape can meet this need by transforming their data model to include a ranking system for the APIs. The ranking system could be based on concrete aspects such as service uptime, licensing compatibility, quality of documentation, and ease of use. Developers who use the APIs can rank them providing real-time feedback to other developers searching for APIs.

Service systems can also increase the engagement between API providers and consumers by giving each API provider the ability to create forums, wikis, or dedicated webpages for their APIs. These features will allow API service providers to create and foster communities around their APIs. API service providers can also showcase successful case studies and recommendations from well-known customers.

Community building is a non-trivial task, and API service providers may not have the capacity to undertake it. Hence, API service systems can help. Tasks such as asking a

developer to rank an API, or ask a service provider to post successful case studies can go a long way in community development, and can be integrated in the features developed by the API service system.

5.5.2 Implications for API service providers

API service providers can benefit from the available features in the existing service system by becoming aware of their communications and fine tuning it. The current programmableweb.com 'submit API' form contains a field that asks "Why is your API different?" that can be filled to communicate aspects of the service provider that makes their API different than others. These aspects should communicate trust and credibility to potential developers. Things like having excellent uptime, having a flexible terms of service, or having a customer with a resonating name matter a lot to potential developers and can differentiate the provider's API.

API providers can make use of the categorization model developed in this study to position their communications among the 6 categories depending on their core strengths. Whether the API provider is communicating enablers, linkers, or experience providers, focusing on features that can be most relevant to target developers such as uptime and service quality for enablers, variety and breadth of connected data and services for linkers, and ease of use and quality of documentation for experience providers can differentiate the provider's API.

5.5.3 Recommendations for entrepreneurs

There is a lot of commercial potential in filling the filters gap by creating new companies

and API service systems that address it. New API service systems should act as supply and demand connectors by utilizing new innovations and technology. Big data and machine learning can play a crucial role in addressing this problem, as well as community building and service system development. Entrepreneurs should become more like service system enablers than service systems.

At the moment, companies addressing this gap are focusing on large enterprise players who have the capacity and the budget to contract expensive services. A potential strategy for incumbents is to focus on smaller companies and startups in specific markets such as Medical Devices, the Internet of Things, or Fintech.

6 Conclusion

APIs have delivered on a promise of having tremendous value within the walls of the enterprise but have fallen short from delivering on the same promise at an internet scale. API service systems are the primary manifestation of APIs at the internet scale. Empirically studying API service systems has the potential of revealing the reason behind their inability to scale APIs at the level of the internet.

Using Topic Modeling to empirically study API descriptions in programmableweb.com, the largest API service system at the moment, a long tail of customer value proposition distributions was revealed. Analyzing the evolution of the tail through time revealed that niche areas are moving into the mainstream and the overall distributions are converging causing the tail to flatten.

The flattening of the tail signals an increase in supply that needs to be met by shifting the demand towards the tail, a need that is unmet at the moment. API service systems can shift the demand towards the tail by acting as filters connecting supply and demand. One example of how they can do that is by allowing API service providers to brand and differentiate their APIs through communicating resonating focus value propositions. To do so, API service systems need to provide features that enables API service providers to create communities and foster them around their APIs. This gap in the industry is up for grasp by aspiring and existing entrepreneurs.

6.1 Lessons learned from applying Topic Modeling

Topic modeling can serve a great value when it comes to analyzing topics that require semantic analysis. Applying Topic Modeling to a corpus of API descriptions allowed the researcher to perform two types of analyses: topic interpretation, and trend analysis. Topic interpretation was based on analyzing the top words associated with each topic. A good model would allow the interpretation to be done through the words only, however, for some topics example documents were required to analyze the topics. The final outcome of the interpretation was the identification and categorization of API customer value propositions. Trend analysis was based on plotting the relevant topic proportions in the documents of the corpus. Topic trend analysis revealed insight into the evolution of API customer value propositions over time and how that affects the API service system as a whole.

The data used in this research were a corpus of API descriptions that were written by API service providers. This limited the study to only draw insights about communicated customer value propositions. In a previous attempt, news articles from programmableweb.com about APIs and how they help API consumers were used. The resulting topics were substantially different, and different analysis could have been done. However, due to the amount of preprocessing required the idea of analyzing news articles was abandoned.

Preprocessing of the data also changes the outcome of the model significantly. The size of the document plays a role in the quality of the outcome. Topic modeling works well with large documents. Small documents such as twitter feeds do not produce good results. API descriptions can be considered small documents, however, they are not as

small as twitter feeds. Having the ultimate judgement being the interpretability of topics, the size of API descriptions did not affect the final outcome. Preprocessing techniques such as stemming, lemmatization, and removal of stop words also play a role in the final outcome. It was found that removing words such as “api” and “service” improved the final outcome and produced topics that are more interpretable.

The number of topics and the hyper-optimization of model parameters affect the final outcome. Mallet has the option of automatically optimizing the LDA parameters during the model training. It was found that keeping this option on produced better results. The number of topics was the highest impact factor in the quality of the outcome. Fewer number of topics produced more general topics and higher number of topics produced more fine-tuned topics.

Using topic modeling requires programming and command line skills that can hinder researchers from applying it in contexts outside of computer science. There is a need for simple and interactive graphical user interface based applications to aid the use of Topic Modeling in areas such as humanities and social science.

6.2 Limitations

This research was limited by two factors, the sources of data used and the analysis technique used. On the data side, the source of data used in this research is secondary and only contains what API service providers say about themselves. Thus, the type of analysis and the conclusions drawn to be only about how providers communicate limiting the depth of the conclusions reached. In addition, due to the large number of

API descriptions obtained, verifying each API description was not possible. This caused the data to contain some descriptions for outdated APIs which may have caused some skewness in the final outcome.

Limitations caused by Topic Modeling as a data analysis technique are mainly due to the fact that Topic Modeling is a machine learning technique and follows the principle of "garbage in garbage out". Analyzing the Topic Model is only as good as what you feed it and can vary tremendously based on the input. Topic Modeling is also known with limitations regarding short documents and API descriptions are considered somewhat short documents. While the final outcome of the topic was interpretable, longer documents could have revealed more abstract topics.

The interpretation of the Topic Model is very subjective and requires a lot of domain expertise to identify gaps in the model or topics that do not make sense. This was remedied by using actual example documents to interpret the topics.

Finally, the LDA algorithm does not take in consideration the dynamism and evolution of the topics. Topic trend analysis was performed after the Topic Model was trained making it only an approximation for how topics evolved over time.

6.3 Future Work

Future research directions can be pursued in using different Topic Modeling algorithms to reveal different types of insight. Other types of Topic Modeling algorithms include

Dynamic Topic Models and Relational Topic Models. Dynamic Topic Models can be used to study the evolution of topics over time instead of constructing approximate trend graphs. While Relational Topic Models could be used along with links between APIs and Mashups to predict links based on content in the API descriptions.

The identified categorization model and trained Topic Model can be used in combination with extra data sources to create a service value map for the API service system. This map can be analyzed to reveal new insight such as the health of the service system, the value gaps in the service system, and answer questions such as how to best create and deliver value in the service system.

The methodology presented here can be used to study other types of service systems and communities such as open source service systems or developer communities. Results from these studies can be used as grounds for comparison between different types of service systems. In addition, lessons learned from more advanced service systems like some of the open source ones could be used guide API service systems.

7 References

Ali, M., & LaPaugh, A. 2013. Enabling Author-Centric Ranking of Web Content. ***International Workshop on the Web and Databases (WebDB)***.

Allee, V. 2008. Value network analysis and value conversion of tangible and intangible assets. ***Journal of Intellectual Capital***, 9(1): 5–24.

Anderson, J. C., Narus, J. A., & Van Rossum, W. 2006. Customer value propositions in business markets. ***Harvard Business Review***, 84(3): 91–99.

Anderson, C. 2006. **Long tail, the, revised and updated edition: Why the future of business is selling less of more**. Hyperion.

Asuncion, H. U., Asuncion, A. U., & Taylor, R. N. 2010. Software traceability with topic modeling. ***2010 ACM/IEEE 32nd International Conference on Software Engineering***, 1: 95–104.

Bachlechner, D., Siorpaes, K., Fensel, D., & Toma, I. 2006. Web service discovery-a reality check. **3rd European Semantic Web Conference**, 308.

Bai, J., Xiao, H., Yang, X., & Zhang, G. 2009. Study on integration technologies of building automation systems based on web services. ***2009 ISECS International Colloquium on Computing, Communication, Control, and Management Study***, 262–266.

Ballantyne, D., Frow, P., Varey, R. J., & Payne, A. 2011. Value propositions as communication practice: Taking a wider view. ***Industrial Marketing Management***, 40(2): 202–210.

Barros, A. P., & Dumas, M. 2006. The rise of web service service systems. ***IT Professional***, 8(5): 31–37.

Beimborn, D., & Joachim, N. 2011. The joint impact of service-oriented architectures and business process management on business process quality: An empirical evaluation and comparison. ***Information Systems and E-Business Management***, 9(3): 333–362.

Beletski, O. 2008. **End user mashup programming environments.** In T-111.5550 Seminar onMultimedia.

Berlind, D. 2016. **ProgrammableWeb's New API Directory Data Model Explained.** http://www.programmableweb.com/news/programmablewebs-new-api-directory-data-model-explained/analysis/2016/07/08. Accessed Sept 2016.

Bizer, C., Heath, T., & Berners-Lee, T. 2009. Linked Data - The Story So Far. **Semantic Services, Interoperability and Web Applications: Emerging Concepts**, 5(3): 205-227.

Blei, D. M., & Lafferty, J. D. 2006a. Dynamic topic models. P**roceedings of the 23rd international conference on Machine learning**, pp. 113-120.

Blei, D., & Lafferty, J. 2006b Correlated topic models. **Advances in neural information processing systems**, 18(147).

Blei, D. M., & Mcauliffe, J. D. 2008. Supervised topic models. ***Advances in Neural Information Processing Systems 22***, 121–128.

Blei, D. M. 2012. Probabilistic Topic Models. ***Communications of the ACM***, 55(4): 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. 2003. Latent Dirichlet Allocation David. ***Journal of Machine Learning Research***, 3: 993–1022.

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S., & Winer, D. 2000. Simple Object Access Protocol (SOAP) 1.1. World Wide Web Consortium note (2000). **http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.** Accessed August 2016.

Bourque, P., & Fairley, R. E. 2014. Guide to the Software Engineering - Body of Knowledge. ***IEEE Computer Society***. www.swebok.org.

Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L. 2005. An approach for QoS-aware service composition on algorithms. ***GECCO 2005 - Genetic and Evolutionary Computation Conference***, 1069–1075.

Carlile, P. R., & Christensen, C. M. 2004. **The Cycles of Theory Building in Management Research.** Working paper.

Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., & Shan, M.-C. 2000. Adaptive and Dynamic Service Composition in eFlow. ***12th International Conference Advanced Information Systems Engineering (CAiSE)***, 3084: 13–31.

Chandler, J. D., & Lusch, R. F. 2015. Service Systems: A Broadened Framework and Research Agenda on Value Propositions, Engagement, and Service Experience. ***Journal of Service Research***, June(1): 1–17.

Chaney, A., & Blei, D. 2012. Visualizing Topic Models. ***Icwsm***, 419–422.

Chang, J., & Blei, D. M. 2009. Relational Topic Models for Document Networks Jonathan. ***Proceedings of the 12th International Confe-rence on Artificial Intelligence and Statistics (AISTATS)***, (PART 1): 81–88.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. 2009. Reading tea leaves: How humans interpret topic models. **Advances in neural information processing systems,** 288-296.

Chang, J. 2015. **R-Package LDA.** CRAN. https://cran.r-project.org/web/packages/lda/lda.pdf. Accessed August 2016.

Chen, X., Lemos, A. L., Barukh, M. C., & Benatallah, B. 2012. Service graph base: A unified graph-based platform for representing and manipulating service artifacts. ***Proceedings - 2012 5th IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2012***, (Section II).

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S., & Rackham, G. 2005. Impact of service orientation at the business level. ***IBM Systems Journal***, 44(4): 653–668.

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. 2001. **Web Services Description Langauge (WSDL) 1.1.** W3C Note 15 March 2001. https://www.w3.org/TR/wsdl. Accessed August 2016.

Chuang, J., Fish, S., Larochelle, D., Li, W. P., & Weiss, R. 2014. Large-Scale Topical Analysis of Multiple Online News Sources with Media Cloud. ***Data Science for News Publishing***.

Clauset, A., Rohilla Shalizi, C., & J Newman, M. E. 2009. Power-Law Distributions in Empirical Data. ***SIAM Review***, 51(4): 661–703.

Crockford, D. 2006. **The application/JSON media type for JavaScript Object Notation (JSON).** IETF Tools. http://www.ietf.org/rfc/rfc4627.txt. Accessed August 2016.

Deerwester, S., Dumais, S. T., Furnas, G. W., & Landauer, T. K. 1990. Indexing by Latent Semantic Analysis. ***Journal of the American Society for Information Science Sep***, 41(6): 391.

Dietz, L., Bickel, S., & Scheffer, T. 2007. Unsupervised prediction of citation influences. **Proceedings of the 24th international conference on Machine learning**, pp. 233-240.

Elfatatry, A. 2007. Dealing with Change: Components Versus Services. ***COMMUNICATIONS OF THE ACM***, 50(8): 35–39.

Elmeleegy, H., Ivan, A., Akkiraju, R., & Goodwin, R. 2008. MashupAdvisor: A recommendation tool for mashup development. ***Proceedings of the IEEE International Conference on Web Services, ICWS 2008***, 337–344.

Endres-Niggemeyer, B. 2013. Semantic Mashups. ***Semantic Mashups***:1-51. Springer Berlin Heidelberg.

Erosheva, E. A. 2002. **Grade of membership and latent structure models with application to disability survey data.** Doctoral dissertation, Office of Population Research, Princeton University.

Erosheva, E., Fienberg, S., & Lafferty, J. 2004. Mixed-membership models of scientific publications. **Proceedings of the National Academy of Sciences**, 101(1):5220-5227.

Evans, P. C., & Basole, R. C. 2016. Economic and Business Dimensions Revealing the API Service system and Enterprise Strategy via Visual Analytics. ***COMMUNICATIONS OF THE ACM***, 59(2).

Fei-Fei, L., & Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.** 2:524-531.

Fielding, R. T. 2000. **Architectural styles and the design of network-based software architectures.** Doctoral dissertation, University of California, Irvine.

Frow, P., McColl-Kennedy, J. R., Hilton, T., Davidson, A., Payne, A., et al. 2014. Value propositions: A service service systems perspective. ***Marketing Theory***, 14(3): 327–351.

Garriga, M., Mateos, C., Flores, A., Cechich, A., & Zunino, A. 2016. RESTful service composition at a glance: A survey. ***Journal of Network and Computer Applications***, 60: 32–53.

Gao, W., Li, P., & Darwish, K. 2012. Joint topic modeling for event summarization across news and social media streams. ***Cikm***, (OCTOBER): 1173.

Gerrish, S., & Blei, D. M. 2010. A language-based approach to measuring scholarly impact. **Proceedings of the 27th International Conference on Machine Learning**, pp. 375-382.

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., et al. 2010. myExperiment: A repository and social network for the sharing of bioinformatics workflows. ***Nucleic Acids Research***, 38(2): 677–682.

Griffiths, T. L., & Steyvers, M. 2002. A probabilistic approach to semantic representation. In **Proceedings of the 24th Annual Conference of the Cognitive Science Society.**

Griffiths, T. L., & Steyvers, M. 2003. Prediction and semantic association. **Neural information processing systems,** (15).

Griffiths, T. L., & Steyvers, M. 2004. Finding scientific topics. **Proceedings of the National Academy of Science**, (101):5228-5235.

Halinen, A., Medlin, C. J., & Tornroos, J.-A. 2012. Time and process in business network research. ***Industrial Marketing Management***, 41(2): 215–223.

Hall, D., Jurafsky, D., & Manning, C. D. 2008. Studying the history of ideas using topic models. **Proceedings of the conference on empirical methods in natural language processing**, pp. 363-371.

Han, Y., Chen, S., & Feng, Z. 2014. Mining Integration Patterns of Programmable Service system with Social Tags. ***Journal of Grid Computing***, 12(2): 265–283.

Hau, T., Ebert, N., Hochstein, A., & Brenner, W. 2008. Where to Start with SOA Criteria for Selecting SOA Projects. ***Proceedings of the 41st Hawaii International Conference on System Sciences***, 1–9.

Hinchcliffe, D., & Benson, J. 2011. **EMML Changes Everything: Profitability, Predictability, & Performance through Enterprise Mashups**. Technical Report. Open Mashup Alliance. http://mdc.jackbe.com/enterprise-mashup/sites/default/files/oma_whitepaper_120309_0.pdf. Accessed August 2016.

Hofmann, T. 1999. Probabilistic latent semantic indexing. ***Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval***, 50–57.

Hofmann, T. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. **Machine Learning Journal**, 42(1):177-196.

Hornik, K., & Grün, B. 2011. topicmodels: An R package for fitting topic models. **Journal of Statistical Software**, 40(13):1-30.

Huang, K., Fan, Y., & Tan, W. 2012. An empirical study of programmable web: A network analysis on a service-mashup system. ***19th International Conference on Web Services***, 552–559. IEEE.

Jacobi, C., van Atteveldt, W., & Welbers, K. 2015. Quantitative analysis of large amounts of journalistic texts using topic modelling. ***Digital Journalism***, 811(November): 1–18.

Jadeja, N., & Pandya, A. 2014. Multi-aspect sentiment analysis with topic models modeling. ***International Journal of Advance Engineering and Research Development***, 1(5).

Koop, D., Schiedegger, C. E., Callahan, S. P., Juliana, F., & Silva, C. T. 2008. VisComplete: Automating Suggestions for Visualization Pipelines. ***IEEE Transactions on Visualization and Computer Graphics***, 14(6): 1691–1698.

Kumar, S., Dakshinamoorthy, V., & Krishnan, M. 2007. SOA and Information Sharing in Supply Chain: "How" Information is Shared Matters! ***Twenty Eighth International Conference on Information Systems***, 21–27.

Lanthaler, M., & Gütl, C. 2010. Towards a RESTful service service system: Perspectives and challenges. ***4th IEEE International Conference on Digital Service systems and Technologies***, 209–214.

Larsen, K. R., & Bong, C. H. 2016. a Tool for Addressing Construct Identity in Literature Reviews and Meta-Analyses. ***MIS Quarterly***, 40(3): 1–23.

Lemos, A. L., Daniel, F., & Benatallah, B. 2015. Web service composition: A survey of techniques and tools. ***ACM Computing Surveys***, 48(3): 33:1-33:41.

Lin, C., & He, Y. 2009. Joint sentiment/topic model for sentiment analysis. **Proceedings of the 18th ACM conference on Information and knowledge management**, 375–384.

Linstead, E., Rigor, P., Bajracharya, S., Lopes, C., & Baldi, P. 2007a. Mining concepts from code with probabilistic topic models. ***Proc. ASE***, (April 2016): 461.

Linstead, E., Rigor, P., Bajracharya, S., Lopes, C., & Baldi, P. 2007b. Mining eclipse developer contributions via author-topic models. ***Proceedings - ICSE 2007 Workshops: Fourth International Workshop on Mining Software Repositories, MSR 2007***.

Linstead, E., Lopes, C., & Baldi, P. 2008. An application of Latent Dirichlet Allocation to analyzing software evolution. ***Proceedings - 7th International Conference on Machine Learning and Applications, ICMLA 2008***, 813–818.

Lu, Y., & Zhai, C. 2008. Opinion integration through semi-supervised topic modeling. ***Proceeding of the 17th international conference on World Wide Web - WWW '08***,

121–130.

Lusch, R. F., & Vargo, S. L. 2006. Service-dominant logic: reactions, reflections and refinements. ***Marketing Theory***, 6(3): 281–288.

Lyu, S., Liu, J., Tang, M., Kang, G., Cao, B., et al. 2014. Three-level views of the web service network: An empirical study based on programmableweb. ***Proceedings - 2014 IEEE International Congress on Big Data, BigData Congress 2014***, 374–381.

Maleshkova, M., Pedrinaci, C., & Domingue, J. 2010. Investigating Web APIs on the world wide Web. ***Proceedings - 8th IEEE European Conference on Web Services, ECOWS 2010***, 107–114.

Manikrao, U. S., & Prabhakar, T. V. 2005. Dynamic selection of web services with recommendation system. **IEEE International Conference on Next Generation Web Services Practices (NWeSP'05)**, pp. 5-pp.

McCallum, A. K. 2002. **Mallet: A machine learning for language toolkit.**

Mccollister, C., Huang, S., & Luo, B. 2015. Building Topic Models to Predict Author Attributes from Twitter Messages. ***CLEF 2015 Labs and Workshops, Notebook Papers***.

Mimno, D., & McCallum, A. 2007. Mining a digital library for influential authors. **Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries**, pp. 105-106.

Minka, T., & Lafferty, J. 2002. Expectation-propagation for the generative aspect model. **Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence,** pp. 352-359.

Muehlen, M. Z., Nickerson J., & Swenson K. 2005. Developing web services choreography standards – the case of REST vs. SOAP. **Decision Support Systems**, 40(1):9–29.

Newman, D., & Chemudugunta, C. 2006. Analyzing Entities and Topics in News Articles. ***International Conference on Intelligence and Security Informatics***, 93–104. Springer Berlin Heidelberg.

Ni, X., Sun, J. T., Hu, J., & Chen, Z. 2009. Mining multilingual topics from wikipedia. **Proceedings of the 18th international conference on World wide web**, pp. 1155-1156.

Nottingham, M., & Syre, R. 2005. **The Atom Syndication Format (RFC 4287)**. Proposed

Standard. IETF Network Working Group. https://tools.ietf.org/html/rfc4287. Accessed August 2016

Paolucci, M., Kawamura, T. . b, Payne, T. R., & Sycara, K. . 2002. Semantic matching of web services capabilities. ***Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)***, 2342 LNCS: 333–347.

Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecky, J., & Domingue, J. 2010. iServe: A linked services publishing platform. **CEUR Workshop Proceedings**, 596.

Pan, W., Chen, S., & Feng, Z. 2012. Investigating the collaborative intention and semantic structure among co-occurring tags using graph theory. ***Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference Workshops, EDOCW 2012***, 190–195.

Pritchard, J. K., Stephens, M., & Donnelly, P. 2000). Inference of population structure using multilocus genotype data. **Genetics**, 155(2):945-959.

Rajasekaran, P., Miller, J., Verma, K., & Sheth, A. 2005. Enhancing Web Services Description and Discovery to Facilitate Composition. ***Semantic Web Services and Web Process Composition: First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers***, 3387: 55–68.

Ramage, D., Rosen, E. 2009. **Stanford Topic Modeling Toolbox.**

Rehurek, R., & Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. ***Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks***, 45–50. Valletta, Malta: ELRA.

Ren, M., & Lyytinen, K. K. 2008. Building enterprise architecture agility and sustenance with SOA. ***Communications of the Association for Information Systems***, 22(January 2008): 75–86.

Rosenblum, H. 2011. ***Customer Values of Communication Enabled Application Mashup Types***. Unpublished master's thesis, Carleton University, Ottawa.

Roy Chowdhury, S., Daniel, F., & Casati, F. 2011. Efficient, interactive recommendation of mashup composition knowledge. ***Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)***, 7084 LNCS: 374–388.

RSS Advisory Board. 2009. **RSS 2.0 Specification.** http://www.rssboard.org/rss-specification. Accessed August 2016.

Singh, M. P. 2001. Physics of service composition. ***IEEE Internet Computing***, 5(June): 6–7.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. 2005. Discovering objects and their location in images. **Tenth IEEE International Conference on Computer Vision,** 1:370-377.

Skalen, P., Gummerus, J., von Koskull, C., & Magnusson, P. R. 2014. Exploring value propositions and service innovation: a service-dominant logic study. ***Journal of the Academy of Marketing Science***, 1–22.

Steyvers, M., & Griffiths, T. 2005. **Matlab Topic Modeling Toolbox.**

Steyvers, M., & Griffiths, T. 2010. Probabilistic Topic Models. ***MIS Quarterly***, 3(3): 993–1022.

Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., & Dustdar, S. 2016. From the Service-Oriented Architecture to the Web API Economy. ***IEEE Internet Computing***, 20(4): 64–68.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. 2006. Hierarchical Dirichlet Processes. ***Journal of the American Statistical Association***, 101(476): 1566–1581.

Titov, I., & McDonald, R. 2008. Modeling online reviews with multi-grain topic models. ***Proceeding of the 17th international conference on World Wide Web (WWW)***.

Torres, R., Tapia, B., & Astudillo, H. X. B. N. 2011. Improving Web API Discovery by Leveraging Social Information. ***2011 IEEE International Conference on Web Services***, 744(1): 744–745.

Varadarajan, J., & Odobez, J. 2009. Topic Models for Scene Analysis and Abnormality Detection.pdf. ***Computer Vision Workshops.***

Vargo, S. L., & Lusch, R. F. 2004**.** Evolving to a new dominant logic for marketing. **Journal of marketing**, 68(1):1-17.

Vargo, S. L., & Lusch, R. F. 2008. Service-dominant logic: continuing the evolution. **Journal of the Academy of marketing Science**, 36(1):1-10.

Wan, Y., Chen, L., Wu, J., & Yu, Q. 2015. Time-Aware API Popularity Prediction via Heterogeneous Features. ***Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015***, 424–431.

Wang, J., Chen, H., & Zhang, Y. 2009. Mining user behavior pattern in mashup community. ***2009 IEEE International Conference on Information Reuse and Integration, IRI 2009***, 126–131.

Wang, C., & Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. ***Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11***, 448–456.

Weinhardt, C., Blau, B., & Conte, T. 2011. **Business aspects of web services**. Springer Science & Business Media.

Weiss, M., & Gangadharan, G. R. 2010. Modeling the mashup service system: Structure and growth. ***R and D Management***, 40(1): 40–49.

Weiss, M., & Sari, S. 2010. Evolution of the mashup service system by copying. ***Proceedings of the 3rd and 4th International Workshop on Web APIs and Services Mashups***, 11:1–11:7.

Weiss, M., Sari, S., & Noori, N. 2013. Niche Formation in the Mashup Service system. ***Technology Innovation Management Review***, 3(5).

Wittern, E., Laredo, J., Vukovic, M., Muthusamy, V., & Slominski, A. 2014. A graph-based data model for API service system insights. ***Proceedings - 2014 IEEE International Conference on Web Services, ICWS 2014***, 41–48.

Woods, D., Brail, G., Jacobson, D. 2011. **APIs: A Strategy Guide**. O'Reilly Media, Inc.

Yu, S., & Woodard, C. J. 2008. Innovation in the programmable web: Characterizing the mashup service system. ***International Conference on Service-Oriented Computing***, 136–147. Springer Berlin Heidelberg.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., et al. 2011. Comparing twitter and traditional media using topic models. ***33rd European Conference on IR Research, ECIR 2011***, 338–349.

Appendix A

A.1 Optimal Topic Model Analysis

5-topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 5-0	
0	service information data access online payment applications services integrate account	Payment Service, Integration, Data access	N	New Topics	5
1	data search service text access information web analysis methods content	Search Service, text analysis	N	Duplicate Topics	0
2	applications service access data management services platform integrate mobile sms	Mobile Integratio, platform management	N	>1 Theme	5
3	data information service access location search xml methods json services	Location Services, JSON data format	N		
4	access applications data information content social service site create integrate	Social content management, data access	N		

10-Topics				
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 10-5

0	applications access integrate functionality include information methods retrieving managing video	Integration		New Topics	5
1	methods data support service information access database search records resources	Database access		Duplicate Topics	0
2	text analysis search content language data web translation image recognition	Text Analysis, Image Recognition		>1 Theme	3
3	sms email messages service send messaging mobile services applications phone	SMS messaging, email services,	N		
4	json site data web responses restful xml calls service formatted	RESTful data format	N		
5	data exchange bitcoin market service information calls json trading account	Cryptocurrency	N		
6	payment online services service payments product products access customers information	Payment services			
7	data applications service platform management services cloud access web json	Cloud access, Platform management	N		
8	service images travel image files applications information file search integrate	File management	N		
9	data information location service access xml map maps locations address	Location services			

15-Topics

Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 15-10	
0	search information product products travel access services offers booking online	Travel Service	N	New Topics	6
1	text analysis language search content translation web data processing recognition	Text Analysis		Duplicate Topics	0
2	json xml responses data calls restful formatted http rest service	Rest data format		>1 Theme	4
3	management data platform service marketing applications services customer software tools	Platform access, marketing	N		
4	data location information service map maps weather time locations access	Location, weather	N		
5	information data access website open public government soap online provided	Open government data	N		
6	methods support service data information access search database submission retrieval	Database access			
7	address service addresses information email domain services number shipping data	Address, email, domain data, shipping data	N		
8	payment payments online service card services credit account processing transactions	Payment services			
9	sms messages send messaging service email mobile services applications phone	SMS Messaging, Email messaging			
10	cloud applications web data access mobile	Cloud access APIs			

	platform services service application				
11	images image files service file documents video upload pdf web	File managament			
12	applications access functionality integrate include methods information retrieving managing create	Integration APIs			
13	content social site access music video information create media share	Social media management, media streaming	N		
14	data exchange bitcoin market trading information currency service trade financial	Cryptocurrency, financial APIs			

20-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 20-15	
0	services address information service website domain email addresses number soap	Address and Domain data		New Topics	4
1	json xml responses restful formatted calls search data service web	REST data format		Duplicate Topics	0
2	data information access database open search research metadata public library	Open data access		>1 Theme	1
3	project management projects job tasks access time online learning system	Project management APIs	N		
4	management email marketing data customer platform service services	Marketing management APIs			

	integrate customers				
5	video music game videos games media content access information audio	Media APIS			
6	applications access functionality integrate methods include retrieving managing information create	Integration APIs			
7	product products shipping information online orders service customers services shopping	Shipping APIs			
8	data location map information maps weather locations code service address	Location APIs			
9	social content site create share twitter information media news web	Social media management			
10	text analysis language content translation data semantic sentiment web sequence	Text analysis			
11	data time monitoring analytics devices performance real-time metrics service reports	Monitoring analytics	N		
12	offers access online services advertising search content deals information platform	Advertising APIs			
13	image images files file text pdf documents service upload format	Image uploading	N		
14	data bitcoin exchange market trading information currency financial trade account	Cryptocurrency			
15	information travel data service booking services access public vehicle search	Travel APIs			

16	applications web cloud services service platform access application mobile apps	Cloud access			
17	json authentication rest key http data requests access service account	Authentication APIs	N		
18	methods service support information data submission retrieval functions specific including	Data access			
19	sms payment messages send service messaging mobile payments services text	Payment APIs, SMS			

25-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 25-20	
0	product products shipping online information services service customers orders shopping	Shipping APIs		New Topics	6
1	social create share events site twitter information people media access	Social media management		Duplicate Topics	0
2	json xml responses calls restful formatted data http rest service	REST data format		>1 Theme	4
3	text analysis language content translation semantic sentiment processing data speech	Text Processing			
4	location information data service locations city code address time weather	Location APIs, Weather data			
5	data exchange market bitcoin trading information currency trade financial price	Cryptocurrency, financial data			

6	data database information sequence protein soap sequences biological vehicle analysis	Protein sequencing			
7	travel information booking search access deals flight hotel offers services	Travel API			
8	game games data information sports players live statistics scores player	Sports statistics			
9	images image maps map recognition photos features face interactive visual	Image processing, maps	N		
10	management software data business services service customer applications platform businesses	Platfrom APIs			
11	data time analytics monitoring reports service performance real-time metrics analysis	Monitoring Analytics			
12	sms messages send messaging email service mobile text phone services	SMS and email messaging			
13	health information data property estate medical real healthcare insurance care	Health information APIs, real estate	N		
14	applications include methods functionality integrate access retrieving managing information create	Integration APIs			
15	search content site information data access web results database news	Search APIs	N		
16	files file service url image documents images create pdf upload	File management			
17	data information access open database government public research project datasets	Open data APIs			

18	methods support service retrieval submission functions including management specific resources	Resource Management	N		
19	video music videos content media audio access information movie streaming	Media streaming			
20	website soap provided calls documentation information xml services english format	SOAP Protocol	N		
21	applications web cloud mobile platform access application service services data	Cloud access APIs			
22	payment payments card online credit service processing transactions account transaction	Payment APIs			
23	address email domain addresses data service services security job numbers	email and domain address authentication			
24	applications access integrate documentation public functionality platform interested mobile social	Mobile Access	N		

30-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 30-25	
0	website information events event soap provided calls documentation english retrieve	SOAP Protocol		New Topics	3
1	location data information map maps city service locations code address	Location APIs		Duplicate Topics	0

2	management services platform applications software service data integrate systems business	Platfrom APIs		>1 Theme	2
3	web site create website code service application javascript free google	Web APIs	N		
4	payment payments online card credit processing transactions account service transaction	Payment APIs			
5	sms messages send email messaging service text mobile message calls	SMS and email messaging			
6	image images files file pdf documents upload text service format	File management			
7	mobile devices app apps applications device android platform application ios	Mobile Access			
8	game games data information sports players scores live statistics player	Sports statistics			
9	xml json responses restful formatted data calls web service returns	REST data format			
10	data weather information energy access conditions location national solar service	Weather			
11	bitcoin service account exchange trading calls currency rest trade information	Cryptocurrency			
12	social content share media twitter news create photos access facebook	Social media management			
13	search access music information database metadata data library books collection	Media search	N		
14	data information access open government public	Open data APIs			

	online learning state education				
15	search offers product access services advertising products online marketing deals	Marketing			
16	json key authentication rest http requests access data format required	Authentication			
17	applications methods include functionality access integrate retrieving managing information create	Information Retrieval			
18	methods service support submission functions information retrieval including specific access	Information Retrieval			
19	cloud storage service data files access servers management server amazon	Cloud access APIs			
20	video content videos media audio live chat streaming service online	Media streaming			
21	text language content analysis translation recognition semantic machine search sentiment	Text Processing			
22	data market financial stock exchange access rates historical quotes information	Financial services API			
23	domain url github domains links urls link service services dns	Domain management	N		
24	data health database information analysis sequence protein medical sequences search	Health information			
25	address data email addresses number numbers phone information	Email address verification			

	verification service				
26	data time analytics reports monitoring service information real-time performance track	Monitoring Analytics			
27	travel information booking data property service search real vehicle services	Travel APIs, Vehicle Information			
28	applications access public documentation functionality integrate interested project service information	Integration APIs			
29	shipping product products service information orders delivery customers print online	Shipping API			

35-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 35-30	
0	social site share twitter create links service url facebook people	Social media management		New Topics	2
1	mobile devices app apps device android application applications access ios	Mobile Access		Duplicate Topics	3
2	travel information booking data service vehicle transit access flight hotel	Travel APIs		>1 Theme	1
3	methods service support functions submission including retrieval information job include	Information Retrieval			
4	json rest key http authentication requests service access offers data	Authentication			
5	sms messages send messaging text mobile	SMS and email messaging			

	service message bulk applications				
6	data information open government access public state xml json u.s	Open data APIs			
7	cloud service storage monitoring access servers github amazon application server	Cloud access APIs			
8	product products online deals shopping orders offers information store ecommerce	E-commerce			
9	payment payments card online credit processing transactions account transaction cards	Payment APIs			
10	text analysis recognition language processing data sentiment content extraction image	Text Processing			
11	marketing data social email analytics campaigns customer media platform customers	Marketing			
12	bitcoin service exchange account calls currency trading information trade orders	Cryptocurrency,			
13	video music videos media content audio streaming movie access service	Media streaming			
14	management tracking project shipping service time projects track tasks software	Porject management, shipping services	N		
15	responses json xml restful formatted calls data service protocol web	REST data format			
16	applications access documentation public integrate functionality interested service	Platfrom APIs			

	information platform				
17	data market exchange financial stock real trading estate property historical	Real estate, financial information			
18	applications include methods functionality integrate access retrieving managing information create	Integration APIs			
19	images image files file service documents pdf upload create document	File management			
20	questions services answer answers captcha yandex project knowledge human life	Question and Answer	N		
21	services applications management provider web service solutions documentation platform business	Platfrom APIs	D		
22	email address addresses phone service domain numbers data number call	Email address verification			
23	game games data sequence protein information database sequences soap players	Sports statistics, protein sequencing			
24	information website soap calls provided service format xml programmatically documentation	SOAP Protocol			
25	location map maps data locations address mapping information places service	Location API			
26	data applications platform web create services access management service tools	Platfrom APIs	D		
27	search information database access results	Search			

	engine site find list data				
28	content news feeds web articles rss feed site blog service	News feeds			
29	data weather information energy access conditions location service database solar	Weather			
30	text translation language word service terms words languages content web	Text Processing	D		
31	code number time information codes return zip data service retrieve	Zip code retrieval			
32	events information health event online learning students medical data education	Health Information			
33	web code javascript php site java google library website html	Web APIs			
34	data access library information research metadata sports database records collection	Database access			

40-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 40-35	
0	property real affiliate estate search listings properties rental website program	Real Estate		New Topics	1
1	social content twitter news media share facebook access data information	Social Media		Duplicate Topics	2
2	domain shipping delivery service services domains tracking mail dns customers	Shipping, Domain services		>1 Theme	1

3	github software development application applications providers range wide integrate repository	Integration API			
4	mobile devices app apps device applications data android platform application	Mobile			
5	cloud data storage service server access servers monitoring amazon management	Cloud			
6	website information soap calls provided format programmatically service xml documentation	SOAP			
7	create management project manage time projects data update access google	Project Management			
8	services applications web platform service integrate content offers access application	Platform			
9	music events event information movie movies shows radio database artists	Media			
10	data ondemand barchart vehicle analysis number xml car request index	Vehicle Information			
11	data travel booking market stock financial access information flight hotel	Travel			
12	health medical information data healthcare care insurance providers fitness drug	Health			
13	marketing email campaigns platform analytics customers social customer advertising media	Marketing			

14	data weather information energy access time conditions location national solar	Weather			
15	methods service support functions submission including retrieval specific information status	Information Retrieval			
16	location map maps information data code address locations service mapping	Location			
17	product products information online shopping deals search access price store	Product infromation			
18	voice call calls phone service applications services chat businesses integrate	Voice and Chat			
19	game games data information sports players live player statistics scores	Sports Statistics			
20	data information open access government state public u.s tax states	Open Data			
21	text language analysis content translation semantic sentiment machine words word	Text Analysis			
22	data management business software platform systems businesses services customer service	Platform	D		
23	payment payments online card credit processing transactions account transaction cards	Payment			
24	code php java python javascript ruby codes net sample languages	Programming Languages			
25	address data email addresses number service	Email Address Verification			

	validation numbers phone verification				
26	video videos content photos media photo audio share sharing upload	Media Sharing			
27	print printing questions answer answers service design webknox knowledge models	Question and answer			
28	data sequence database protein sequences analysis biological soap research genes	Protein sequencing			
29	responses json xml formatted restful calls data service protocol web	REST			
30	data information traffic transit routes times public service bus time	Transit and Traffic information			
31	job learning students jobs information university student school system education	Learning management systems	N		
32	web site website url service page javascript websites link links	URL Management			
33	project rest bot bots platform tools life projects goals management	Project Management	D		
34	images image files file pdf documents upload recognition document conversion	File Management			
35	bitcoin exchange trading service account trade currency calls orders information	Cryptocurrency			
36	search data access database information results metadata library query research	Search API			
37	json key http rest authentication requests	Authentication			

	service access data account				
38	sms messages send messaging text mobile message bulk sending service	SMS messaging			
39	applications access functionality integrate include methods information retrieving managing create	Integration API			

45-Topics					
Topic ID	Words	General Interpretation	N-New D-Duplicate	Delta 45-40	
0	address domain email addresses number data numbers validation phone verification	Email address Verification		New Topics	0
1	applications include methods integrate functionality access retrieving managing information create	Integration API		Duplicate Topics	6
2	share photos create social service photo site sharing twitter information	Social Media Management		>1 Theme	0
3	access information data library research learning resources metadata online university	Library access			
4	events event information tickets calendar ticket online family venues listings	Events listings			
5	web site website javascript google code create applications application websites	Programming languages			

6	social media data network platform twitter facebook networks offers engagement	Social Media Management	D		
7	data analysis database sequence protein sequences information soap search biological	Protein sequencing			
8	product products online information shopping deals store items price stores	Product Information			
9	data time service information monitoring reports real performance track analytics	Monitoring			
10	data weather map maps energy mapping location information access conditions	Maps, weather			
11	analysis text recognition content language sentiment machine speech detection learning	Text Analysis			
12	data platform exposes analytics sources returns restful analysis information tools	Analytics			
13	questions answer answers technology webknox knowledge question services web base	Question and Answer			
14	information data vehicle transit traffic route routes service public bus	Traffic information			
15	video videos content media audio streaming platform live service files	Media streaming			
16	email marketing campaigns customer campaign management manage emails contact sales	Marketing			
17	travel booking search property information real	Travel			

	estate flight hotel hotels				
18	voice call calls phone chat live messaging service communications applications	Voice chat			
19	text language words semantic word terms analysis content web extraction	Text Analysis	D		
20	content news feeds articles feed rss search books access reviews	Content			
21	methods service support including submission functions retrieval specific access data	Information retrieval			
22	bitcoin exchange calls account trading service currency orders rest trade	Cryptocurrency			
23	sms messages send messaging mobile text bulk message service sending	SMS Messaging			
24	responses json xml formatted restful calls data web protocol service	REST			
25	mobile devices app apps device android application applications ios access	Mobile			
26	services applications management service platform integrate web access software systems	Platform			
27	information tax u.s access companies online business database organizations sales	Database access Taxes			
28	data open information government access public state city united datasets	Open Data			
29	cloud storage service servers amazon files hosting server access file	Cloud			

30	payment payments card credit online processing transactions account transaction service	Payment			
31	url urls service links link text simple short number returns	URL and link management			
32	website information calls soap provided programmatically format service documentation xml	SOAP			
33	image images files file pdf documents translation upload service conversion	File Management			
34	shipping service delivery print printing tracking mail customers orders rates	Shipping			
35	health information data medical food healthcare fitness care drug recipes	Health			
36	search web engine site results service services find keyword sites	Search			
37	game games data players sports information live player statistics league	Sports statistics			
38	data market financial exchange stock rates trading historical barchart markets	Financial API			
39	json key rest http requests authentication access data offers format	Authentication			
40	location information code address data locations city service local places	Location	D		
41	access applications documentation public integrate functionality interested service information mobile	Mobile Integration	D		

42	github application authentication security range software secure wide development service	Github	D		
43	music access movie movies database radio artists information search shows	Media information	D		
44	project management projects job tasks time jobs tracking create team	Project Management			

A.2 Topic Model 40-Topics Diagnostics

Topic ID	Label	tokens	Document Entropy	Coherence	Rank 1 Docs
0	1	4145	6.077	-132.6077	0.2282
1	2	15515	7.4201	-98.1263	0.1802
2	3	6349	6.2316	-129.8569	0.264
3	4	3637	5.9617	-56.8569	0.2413
4	5	12009	7.2243	-99.9886	0.1683
5	6	13614	7.1269	-103.7207	0.1939
6	7	9789	6.997	-66.9379	0.1529
7	8	15917	7.4878	-101.0618	0.1481
8	9	45475	8.5506	-101.6913	0.1345
9	10	7096	6.4416	-142.3019	0.3019
10	11	4534	5.5621	-132.3278	0.2987
11	12	11110	6.7242	-145.894	0.2456
12	13	4050	5.9291	-129.3984	0.2393
13	14	14725	7.2599	-92.569	0.2101
14	15	9529	6.4665	-94.4813	0.2601
15	16	22807	7.5628	-91.5498	0.1111
16	17	12392	6.9986	-88.0853	0.21
17	18	12214	7.0431	-100.8506	0.2179
18	19	7164	6.4441	-83.9634	0.2729
19	20	6782	6.1877	-85.7157	0.3606
20	21	13893	7.0489	-104.9628	0.201
21	22	12353	6.8748	-106.6158	0.2149
22	23	24303	7.8061	-92.1162	0.1621
23	24	11529	6.7254	-70.2148	0.2724

24	25	5854	6.5834	-116.1633	0.1404
25	26	10358	6.7873	-81.3456	0.2013
26	27	7606	6.6909	-101.505	0.2127
27	28	3858	5.8362	-198.3161	0.1832
28	29	6632	5.9591	-86.3229	0.361
29	30	21155	7.9959	-72.0615	0.1013
30	31	7268	6.4073	-87.9058	0.2542
31	32	5808	6.2887	-137.9151	0.2266
32	33	20978	7.9032	-116.6143	0.1153
33	34	4315	5.9665	-133.9748	0.1672
34	35	8816	6.8051	-105.164	0.1836
35	36	8603	6.2958	-64.0209	0.3568
36	37	21018	7.6867	-111.524	0.1379
37	38	29093	8.2032	-83.9838	0.1231
38	39	14366	6.8734	-62.8485	0.2889
39	40	23441	7.7624	-44.2949	0.1924

A.3 Sample from output-doc-topic file for 40-topics

Each line in the file contains the document number, the name of the document represented as a path to the file containing the document, and 3 pairs of topic – distribution numbers separated by tabs. The name of the file containing the document is the API name, and the document in the file is the API description.

7110 file:/mallet/mallet-2.0.8RC3/../api_descriptions/Microsoft_Telephony.txt 24 0.361021188908547 4 0.32205510967179407 8 0.08939318113951183

7111 file:/mallet/mallet-2.0.8RC3/../api_descriptions/Microsoft_Translator.txt 32 0.5467361041068743 21 0.23398216053413015 37 0.08831377696044491

7112 file:/mallet/mallet-2.0.8RC3/../api_descriptions/MIDAS.txt 6
0.29056900900133376 22 0.23956872837545545 8
0.1377578133880396

7113 file:/mallet/mallet-2.0.8RC3/../api_descriptions/Middlecoin.txt 35
0.48548280027362317 13 0.18348776653375007 15
0.15419521295222544

7114 file:/mallet/mallet-2.0.8RC3/../api_descriptions/midíadía.txt 37
0.36484877851528835 17 0.27752345081644714 8
0.1964841978824068

7115 file:/mallet/mallet-2.0.8RC3/../api_descriptions/Mifos.txt 22
0.6822611822272877 29 0.14667836669011772 5
0.07329380253827886

7116 file:/mallet/mallet-2.0.8RC3/../api_descriptions/Mighty_Slider.txt 26
0.30095178580814495 15 0.23627955550906288 37
0.13826818028384297

7117 file:/mallet/mallet-2.0.8RC3/../api_descriptions/MightyCast_NEX.txt 4
0.6089708653636292 36 0.14614056954002344 22
0.11084637484406136

7118 file:/mallet/mallet-2.0.8RC3/../api_descriptions/migme.txt 13
0.31233354211121755 8 0.22743858718104373 1

0.22370888433032893

7119 file:/mallet/mallet-2.0.8RC3/../api_descriptions/miiCard.txt 23

0.38333173616260723 9 0.29689633941308178 0.2716090185941814

7120 file:/mallet/mallet-2.0.8RC3/../api_descriptions/MileSplit.txt 19

0.36889336424614916 1 0.15965639461454337 37

0.10915960634007972

7121 file:/mallet/mallet-2.0.8RC3/../api_descriptions/milkySMS.txt 38

0.9108976099769645 8 0.01067648269621506

A.4 Example API description documents for each topic

API Name	Topic ID	Topic Distribution
Quag	2	0.802216987
Seesmic	2	0.806650663
ShareThis_WebShare	2	0.808596946
Bring_Easy_Return_Service	3	0.806100468
Bring_Tracking	3	0.818776009
USPS_Hold_for_Pickup	3	0.889323571
USPS_International_Shipping_Labels	3	0.860920462
USPS_Online_Express_Mail_Label	3	0.878526048
USPS_Package_Pickup	3	0.873083121
USPS_Signature_Confirmation_Label	3	0.917073791
USPS_Track_&_Confirm	3	0.822896863
GitHub_Activity_Starring	4	0.88052261
GitHub_Activity_Watching	4	0.808196693
GitHub_Emojis	4	0.900652583
GitHub_Gists_Comments	4	0.844159055
GitHub_Git_Blobs	4	0.833769893
GitHub_Git_Commits	4	0.885117775
GitHub_Git_Trees	4	0.931483823
GitHub_Issue_Comments	4	0.833769893

GitHub_Issue_Milestones	4	0.867102477
GitHub_Issues	4	0.833769893
GitHub_Repository_Commits	4	0.834060921
GitHub_Users	4	0.813319411
AppLamp_Wifi_LED	5	0.868366976
Erizo	5	0.915742209
Prowl	5	0.8156903
Atlantic	6	0.820532841
Google_Cloud_Storage	6	0.817150652
Google_Compute_Engine_Instance_Group_Manager	6	0.834769803
HP_Cloud_Compute	6	0.828335532
ilandcloud	6	0.872635076
MuninMX	6	0.801724506
Rackspace_Cloud_Big_Data	6	0.809681916
Rackspace_Cloud_Metrics	6	0.814116463
Rackspace_Cloud_Networks	6	0.823275737
Rackspace_Cloud_Orchestration	6	0.828335532
RhoConnect	6	0.83505709
RightScale_Cloud_Analytics	6	0.825050572
RightScale_Self-Service	6	0.825050572
ShepHertz_App42_Cloud_NoSQL_Storage	6	0.865985993
ShepHertz_App42_Cloud_Session_Management	6	0.844826637
ShepHertz_App42_Cloud_User_Management	6	0.870112604
Songsterr	10	0.847826089
Ticketmaster_International_Discovery	10	0.806160202
AgriCharts_getChart	11	0.840414259
AgriCharts_getFuturesExpirations	11	0.88946241
AgriCharts_getFuturesOptions	11	0.895279972
AgriCharts_getFuturesOptionsExpirations	11	0.88946241
AgriCharts_getInstrumentDefinition	11	0.87910092
AgriCharts_getSignal	11	0.821798894
AgriCharts_getSpecialOptions	11	0.842650261
Australia_Car_Registration	11	0.912067886
Car_Registrations_in_Portugal	11	0.912067886
Dutch_Car_License_Plate_Lookup	11	0.865144319
Finnish_Vehicle_Registration_Searches	11	0.857438428
getChart	11	0.919197387
getEquityOptions	11	0.871301149
getFuturesExpirations	11	0.933676452

getFuturesOptionsExpirations	11	0.933676452
getFuturesSpecifications	11	0.886009306
getHighLows	11	0.818496098
getIndexMembers	11	0.92893917
getInstrumentDefintion	11	0.915112895
getMomentum	11	0.892170853
getQuoteEOD	11	0.914580184
getRatings	11	0.868693089
getScreener	11	0.81109125
getSignal	11	0.900257903
getSpecialOptions	11	0.867011284
HPE_Haven_OnDemand_Connector_History	11	0.838117784
HPE_Haven_OnDemand_Connector_Status	11	0.921323734
HPE_Haven_OnDemand_Delete_Connector	11	0.839043787
HPE_Haven_OnDemand_Delete_Text_Index	11	0.824743281
HPE_Haven_OnDemand_List_Indexes	11	0.811516527
HPE_Haven_OnDemand_Retrieve_Config	11	0.860705646
HPE_Haven_OnDemand_Retrieve_Index_Fields	11	0.95025706
HPE_Haven_OnDemand_Start_Connector	11	0.829521069
HPE_Haven_OnDemand_Update_Connector	11	0.868693089
Norwegian_Car_Registration_Lookup	11	0.853245524
Search_French_License_Plates	11	0.886009306
Swedish_Car_Registration_Search	11	0.83362114
UK_&_Irish_Car_Registration_Lookups	11	0.943151062
Verifica_Targhe_Italiane	11	0.889175655
Hotelbeds_Apitude_Content	12	0.865130297
Mergent_Company_Fundamentals	12	0.883319243
Xignite_Get_Real_Time_Rate	12	0.828720383
Xignite_GetHistoricalRates	12	0.940388357
Xignite_Global_Indices_Historical	12	0.901094743
Xignite_GlobalHistoricalFile	12	0.858316826
Xignite_Initial_Public_Offering_Calendar_&_Performance_Data	12	0.834246871
Xignite_Realtime	12	0.945355901
XigniteAnalysts	12	0.912730425
XigniteBATSRealTime	12	0.948231855
XigniteBondsRealTime	12	0.848682895
XigniteEstimates	12	0.905543792
XigniteFutures	12	0.900821062

XigniteGlobalBondMaster	12	0.877053656
XigniteGlobalExchanges	12	0.834429575
XigniteGlobalMaster	12	0.830070492
XigniteGlobalOptions	12	0.829511159
XigniteGlobalQuotes	12	0.913757255
XigniteGlobalRealTime	12	0.917578643
XigniteGlobalRealTimeOptions	12	0.901094743
XigniteMoneyMarkets	12	0.84854082
AdRout	14	0.816839597
TM_Forum_Customer_Management	14	0.835155226
TM_Forum_Performance_Management	14	0.858704026
TM_Forum_Quote	14	0.829471055
TM_Forum_SLA_Management	14	0.845457807
CDYNE_Weather	15	0.847174725
CORDC_COAMPS_Winds_Model	15	0.877658286
NASA's_Asterank	15	0.822377128
NASA_Fireball_And_Bolide_Reports	15	0.903085491
NCEP_Forecast	15	0.845285244
NSIDC	15	0.85777466
FraudLabs_ZIPCodeWorld_United_States	17	0.932527253
Geobytes_Get_City_Details	17	0.801239397
Geobytes_Get_Distance	17	0.822977294
GeoBytes_Get_Nearby_Cities	17	0.829081383
Map_Data_Services_QuickMap	17	0.803989788
Macy's_Catalog_and_Store_Services	18	0.818836926
Macy's_Shopping_Bag_Services	18	0.820864715
Amadeus_Rail-Station_Autocomplete	19	0.885475206
Nexmo_VoiceXML	19	0.858208274
Plivo_Call	19	0.8222285
Plivo_Call_Play	19	0.834266246
Plivo_Call_Request	19	0.851118926
Plivo_Call_Speak	19	0.870537627
Plivo_Conference	19	0.851118926
Halo_Profile	20	0.824567089
Halo_Stats	20	0.834312997
Italy_SerieA_League_Live_and_Historical_Results	20	0.852504506
Live_Scoreboards	20	0.852504506
Riot_Games	20	0.83392114
Roanuz_Cricket	20	0.906799533

Roanuz_Cricket_Authentication	20	0.893485501
Roanuz_Cricket_Football_Match	20	0.875733583
Roanuz_Cricket_ISL_Football	20	0.870330854
Roanuz_Cricket_ISL_Football_Season	20	0.857981808
Roanuz_Cricket_Match	20	0.919393961
Roanuz_Cricket_Match_Over_Summary	20	0.80044142
Roanuz_Cricket_News_Aggregation	20	0.943358868
Roanuz_Cricket_Player_Stats	20	0.909623732
Roanuz_Cricket_Recent_Matches	20	0.895199812
Roanuz_Cricket_Recent_Seasons	20	0.947830457
Roanuz_Cricket_Schedule	20	0.875550191
Roanuz_Cricket_Season	20	0.91478799
Roanuz_Cricket_Season_Points	20	0.936050494
Roanuz_Cricket_Season_Team	20	0.939926143
Sportradar_Cricket	20	0.864436983
Sportradar_Golf	20	0.871535769
Sportradar_MLB	20	0.886217112
Sportradar_NBA	20	0.941692986
Sportradar_NCAA_Football	20	0.949168115
Sportradar_NCAA_Men's_Basketball	20	0.923527532
Sportradar_NCAA_Women's_Basketball	20	0.871535769
Sportradar_NFL	20	0.857562339
Sportradar_NHL	20	0.864436983
Sportradar_Odds	20	0.886217112
Sportradar_Olympics	20	0.867253724
Sportradar_Soccer	20	0.84430085
Sportradar_WNBA	20	0.944932204
World_Cup_in_JSON	20	0.839278406
BusinessUSA_Events	21	0.851454661
BusinessUSA_Programs	21	0.884745505
CKAN_Czech_Republic	21	0.830797367
FedSpending	21	0.801357031
GuideStar_Detail	21	0.834652046
International_Aid_Transparency_Initiative	21	0.917654535
It's_Your_Parliament_EU_Data	21	0.801899829
Larimer_County_Public_Records_Databases	21	0.871628204
National_Crime_Victimization_Survey	21	0.800863571
New_York_Times_Congress	21	0.801899829
New_York_Times_NY_State_Legislature	21	0.835970708

OffeneDaten	21	0.860481688
Open_State_Project	21	0.801357031
Open_States	21	0.87648267
OpenColorado	21	0.823124336
AlchemyAPI__Language_Detection	22	0.847796585
AlchemyAPI_Concept_Tagging	22	0.816021162
AlchemyAPI_Entity_Extraction	22	0.840551453
AlchemyAPI_Keyword_Extraction	22	0.849881544
Idilia_Language_Graph	22	0.8583242
Proxem_Ontology-Based_Topic_Detection	22	0.859111736
Allied_Wallet	24	0.858712305
Bitcoin_Payment	24	0.952021841
Darkcoin_Payment	24	0.878852313
Dogecoin_Payment	24	0.952021841
DoneCard	24	0.853072674
Feathercoin_Payment	24	0.927631998
Openpay	24	0.814720653
Potcoin_Payment	24	0.903242155
Reddcoin_Payment	24	0.903242155
Speedcoin_Payment	24	0.927631998
Vericoin_Payment	24	0.952021841
Vertcoin_Payment	24	0.952021841
BankVal_International	26	0.912829418
Data8_Postcode_Lookup	26	0.805762742
FraudLabs_MailBox_Validator	26	0.828836444
FullContact_Disposable_Email	26	0.801036446
WebKnox_Jokes	28	0.846358101
WebKnox_Proxies	28	0.861742556
WebKnox_Question-Answering	28	0.909291426
WebKnox_Words	28	0.800204733
Center_for_Biological_Sequence_Analysis	29	0.863027226
ChromA	29	0.840576147
CSC_PairsDB	29	0.821943437
EBI_ClustalW2_Phylogeny	29	0.912186923
EBI_Lalign	29	0.893370288
EBI_NCBI_BLAST	29	0.893370288
EBI_T-Coffee	29	0.805229939
EMBOSS_Matcher	29	0.879223564
EMBOSS_Needle	29	0.917065336

EMBOSS_Stretcher	29	0.944842593
EMBOSS_Water	29	0.861510821
IBCP_gBIO	29	0.865253703
MyHits	29	0.906698721
Nuclear_Protein_Database	29	0.846523068
Phylogenetic_Footprinting	29	0.808247134
PlantTFDB	29	0.822542574
Rose	29	0.857554063
TheBus	31	0.808559935
Trafiklab_SL_Real_Time_3	31	0.833959477
Trafiklab_SL_Trip_Planner_2	31	0.818865193
NAOqi_DCM	32	0.815522556
Pemilu_FAQ_Presiden	32	0.914829773
Atlassian_Stash_Audit_Rest	34	0.825186823
Atlassian_Stash_Branch_Permissions_Rest	34	0.916873388
Atlassian_Stash_Branch_Utilities_REST	34	0.887533856
Atlassian_Stash_Build_Integration_Rest	34	0.916873388
Atlassian_Stash_Comment_Likes_REST	34	0.929786975
Atlassian_Stash_Core_Rest	34	0.833524874
Atlassian_Stash_JIRA_Integration	34	0.918012099
Atlassian_Stash_SSH_Rest	34	0.897626884
Gupshup_HipChat_Bot	34	0.889440832
Gupshup_InApp_Bot	34	0.926476944
Gupshup_Slack_Bot	34	0.926476944
Gupshup_Teamchat_Bot	34	0.889440832
Gupshup_Twitter_Bot	34	0.815368608
Gupshup_WeChat_Bot	34	0.926476944
ConvertAPI_Email2Pdf	35	0.811722496
ConvertAPI_Excel2Pdf	35	0.923939914
ConvertAPI_Image2Pdf	35	0.83426992
ConvertAPI_Jnt2Pdf	35	0.917601692
ConvertAPI_Lotus2Pdf	35	0.869307724
ConvertAPI_OpenOffice2Pdf	35	0.942166011
ConvertAPI_Pdf2Image	35	0.936569295
ConvertAPI_Pdf2PowerPoint	35	0.936569295
ConvertAPI_PostScript2Pdf	35	0.9152475
ConvertAPI_PowerPoint2Pdf	35	0.87265875
ConvertAPI_Project2Pdf	35	0.886676622
ConvertAPI_Publisher2Pdf	35	0.883343648

ConvertAPI_RichText2Pdf	35	0.9152475
ConvertAPI_Snp2Pdf	35	0.862047178
ConvertAPI_Text2Pdf	35	0.886676622
ConvertAPI_Visio2Pdf	35	0.943818378
50BTC	36	0.838397146
796_Xchange	36	0.850517155
Bitcurex	36	0.861685519
BitKonan	36	0.84041386
BitMarket	36	0.906895256
bitNZ	36	0.807119854
BTC_to_X	36	0.862781998
Bter	36	0.906895256
CaptchaCoin	36	0.920825552
Coingaia_Trade	36	0.814435129
CoinRelay	36	0.84082771
Coins-E	36	0.846949562
Comkort	36	0.860946027
Cryptank	36	0.854163018
Crypto-Trade	36	0.886833221
Cryptonator	36	0.819123063
emeBTC	36	0.853549534
EtherScan_General_Stats	36	0.875861213
EtherScan_Geth_Proxy	36	0.819123063
Evergreen	36	0.806131004
Feathercoin	36	0.838397146
ICBIT_Trading	36	0.805383333
itBit_Trading	36	0.800517998
Litecoin-invest	36	0.886304631
Poloniex	36	0.870014632
Tagbond	36	0.919476749
TheRockTrading	36	0.870014632
Twenty15Coin	36	0.810016989
Vicurex	36	0.882960716
VirCurEx	36	0.857635272
247-BulkSMS	39	0.819388156
24X7SMS	39	0.869467773
5star_SMS	39	0.9183226
AT&T_SMS	39	0.824700163
BulkSMSVilla	39	0.936765483

Clickatell_Connect	39	0.830939648
Clickatell_FTP	39	0.893567127
Clickatell_SMTP	39	0.815552069
Club080	39	0.834659961
ComposeSMS	39	0.861390251
eStore_Bulk_SMS	39	0.934657713
Everlanka_Advertising	39	0.805849226
excelngSMS	39	0.876657105
FDLINK_SMS	39	0.894277094
Freesms	39	0.93874152
GenesisBulkSMS	39	0.890361529
GiftedSMS	39	0.881590687
GlobalBulkSMS	39	0.807736032
GSMA_OneAPI_MMS	39	0.812071964
Intelli_Messaging	39	0.850994696
LinkmySMS	39	0.872250758
Mayorfirst_SMS	39	0.801327377
MessageBird_Voice_Messaging	39	0.801327377
milkySMS	39	0.91089761
MireZone_SMS	39	0.837087222
Mobyt_SMS	39	0.942344889
MyGateSMS	39	0.9405978
NetBulkSMS	39	0.886144774
OneAPI4SMS	39	0.828958713
OVERTURES_SMS	39	0.8418135
Pearl_SMS	39	0.828958713
QUICK_SMS	39	0.849690567
Recharge_Blast	39	0.858563668
RedSMS	39	0.869467773
SFW_SMS	39	0.934657713
Skycore	39	0.854111345
SMSBump	39	0.886144774
SmsDial	39	0.915421268
SMSWebGh	39	0.889993376
Softnet_SMS	39	0.816289305
SplendidSMS	39	0.93874152
SurfBulkSMS	39	0.841591767
Telstra_SMS	39	0.897922627
TripleClick_SMS	39	0.897922627

Universal_SMS_Advertising	39	0.8418135
Utexta_Bulk_SMS	39	0.801592846
Vectramind	39	0.813744155
Vizz_Media	39	0.801137756
Vodacom_Bulk_SMS_Messaging	39	0.897922627
way2easy	39	0.844993496

A.5 Model interpretation and analysis

Topic1: {property real affiliate estate search listings properties rental website program}

The topic contains words that describe real estate, properties, listings and searching. Example APIs descriptions with this topic are Tripping, Rio Branco Real Estate, and Zillow Home Valuations. Tripping is a vacation rental search utility that allows users to search for short-term vacation rental homes in multiple locations at once. The Tripping API gives developers a programmatic interface to the same utility allowing them to search for properties, or retrieve information about a given listing. Rio Branco Real Estate is a Brazilian Real Estate company. Its API gives users the ability to search the site for available properties. Zillow Home Valuations is a real estate network where member sites can each be a real estate portal. The API offers developers the ability to search for home valuation results, charts, comparable houses, market trend charts, property level information, city and neighborhood market statistics, and mortgage rates and monthly payment estimates.

The three APIs allow developers to create applications that can access real estate data without actually collecting it themselves. However, Zillow Home take it a step further by allowing developers to become real estate portals themselves. This makes the purpose

of APIs described by topic 1 is to search for Real Estate Information. The benefits to developers is access to data and information they previously could not have access to, and that is expensive to get.

Topic 2: {social content twitter news media share facebook access data information}

The topic contains words that describe access to social media data and information as well as sharing of content. Examples of API descriptions mainly composed from this topic include Seesmic and ShareThis Webshare.

Seesmic is social media management tool that was acquired by Hootsuite in 2012[5]. The description of the Seesmic API says that it allows its consumers to view who a user follows, and get information about the content the user posts. ShareThis is a tool that allows users to share to multiple social media sites at once. Its WebShare API allows developers to integrate ShareThis's functionality into their apps so that they are able to create applications that can connect to multiple social media sites at the same time.

The main purpose of APIs described by these documents is provide developers with the ability to share content on multiple social media sites at the same time. The main benefits to the developers is they do not have to go through the trouble of implementing integrations to multiple social media sites at the same time. This reduces their implementation cost and increases their connectivity.

Topic 3: {domain shipping delivery service services domains tracking mail dns

---

[5] https://www.crunchbase.com/organization/seesmic#/entity

customers}

Topic 3 descibes shipping and delivery tracking services. The topic also contains the word “domain” and “dns”, but the main theme of the topic is shipping services. This is evident by the API descriptions that are mostly composed of topic 3 which include Bring Easy Return Service, Bring Tracking, and 6 USPS APIs.

Bring Cargo is a Norwegian logistics and transportation services company. Their API allows developers to integrate into their applications Bring’s services. Thereby enabling the developers’ customers to get shipping information directly. USPS is also a major US postal services company. Their suite of APIs can be used by developers to integrate into their applications the ability to track packages, get information about packages, perform scheduling tasks, and more.

APIs described by this topic allow developers to integrate shipping and delivery services provided by major logistics and postal services companies into their applications. The benefit these APIs gives to developers is the ability to provide shipping and delivery services without being a shipping and delivery company.

Topic 4: {github software development application applications providers range wide integrate repository}

This topic is very specific and describes GitHub APIs and APIs that are similar to Github. The APIs allow developers to integrate software repository management functionality into their applications. It also allows the programmatic management of software repositories for the purposes of automation. The main benefit to API consumers is to

use software repository management functionality without building infrastructure.

Topic 5: {mobile devices app apps device applications data android platform application}

Topic 5 is centered around mobile applications and platforms. Three API description examples give more insight into the topic: AppLamp Wifi LED, Erizo, and Prowl. AppLamp Wifi LED API allows customers of the AppLamp to write their own applications to control the Wifi LED lamp. Erizo API allows developers to integrate their own scripts and applications to use the videoconferencing technology provided by Lynckia's Licode WebRTC platform. Prowl API allows users to write scripts for Growl's iPhone client application.

The examples show that the main purpose of APIs described by this topic is to integrate mobile applications with the main platform provided by the service provider. The main benefit to developers is extending existing applications or building their own allowing them to personalize and customize the use of the products.

Topic 6: {cloud data storage service server access servers monitoring amazon management}

Topic 6 describes access to, monitoring and management of cloud services. Example APIs for this topic include Google Cloud Storage, IlandCloud, and Rackspace Cloud Orchestration. The main purpose of the APIs as described in the example documents is to allow programmatic access to the cloud infrastructure services provided by the service provider. The benefit to developers is allowing them to access cloud resources without acquiring the infrastructure in house.

Topic 10: {music events event information movie movies shows radio database artists}

This topic describes databases for media information such as music, movie, events, artists, and radio shows. Two API description examples for this topic are Songsterr, and Ticketmaster International Discovery. Songsterr allows developers to search for songs in their database. Ticketmaster API allows developers to programmatically access Ticketmaster's database of tickets, concerts, sporting events, plays, fairs, and more.

The main purpose of APIs described by this topic is to give programmatic access and searching capability to databases owned by companies such an online ticket retailer. The benefits to developers is gaining access to information they could not get access to previously.

Topic 11: {data ondemand barchart vehicle analysis number xml car request index}

This topic focus on access of vehicle information. Examples in this topic include Australia Car Registration API, Car Registration in Portugal, and Finnish Vehicle Registration Authority. The main purpose of APIs described by these APIs is to provide developers access to car registration information from multiple countries. The main benefit to developers is having access to data they could not get access to before as well as querying multiple sources of data at once.

Topic 12: {data travel booking market stock financial access information flight hotel}

Topic 12 is the topic that has two themes. The two themes in this topic are travel and financial information. From the words in the topic, the travel theme revolves around

booking and specifies flights and hotels as entities. The financial theme focuses on market, stock, and access.

Examples of travel API descriptions in this topic include the suite of Hotelbeds Aptitude APIs. Examples of the financial APIs include Mergent Company Fundamentals, and Xignite suite of APIs.

Hotelbeds Aptitude API provides developers with access to Hotelbeds's database of hotels. The database is Hotelbeds's main product provided to B2B partners. The value to API consumers in this case is getting access to data that is hard to collect and that they do not have access to before.

Mergent Company Fundamental APIs and Xignite's suite of APIs provide financial information such as historical financial statements of publicaly traded companies, information on executives, currency conversion rates, and current economic events. Mergent's APIs focuses on crude data collected over history, while Xignite's APIs provide access to processed data such as market statistics and trends. Xignite's APIs also provide access to real-time financial data.

Both financial APIs have the purpose of giving developers access to financial data gathered, aggregated, and analyzed by the provider. The main benefit to the developers is having access to hard to collect data.

Topic 13: {health medical information data healthcare care insurance providers fitness drug}

Topic 13 describes the general field of health care data. Examples of APIs in this topic are Rx Aggregation API, Doctoralia, and GETHealth API. Rx Aggregation API and Doctoralia API allows developers to access the providers aggregated and collected databases of prescriptions and healthcare professionals. GETHealth API allows developers to integrate their Applications with the myriad of wearable devices such as Fitbit, Jawbone and Lifefitness. The main benefit from both types of APIs is allowing developers to access multiple resources at once such as aggregated data, or multivendor wearable devices.

Topic 14: {marketing email campaigns platform analytics customers social customer advertising media}

Topic 14 describes marketing and advertising analytics. Examples of API descriptions in this topic are AdRout and TM Forum APIs. AdRout API gives developers access to traffic quality scoring and attribution metrics of ads provided by AdRout. The main service allows users to optimize and improve their marketing budget, and the API allows users to retrieve continuous reports on their campaigns. The TM Forum suite of APIs give developers access to the TM Forum's platform. TM Forum provides its members the ability to co-create, prototype, deliver, and monetize digital services for their customers. Their APIs provide developers the ability to manage SLA lifecycle, manage customers, and manage performance over standardized interfaces. The value to developers using these types of APIs is getting access to functionality that is costly to develop in terms of infrastructure and know how.

Topic 15: {data weather information energy access time conditions location national solar}

This topic describes weather and energy related information. Examples of API descriptions in this topic include CDYNE Weather API, NCEP Forecast API, and NSIDC API. The three APIs provide developers with access to weather related data such as Earth's cryosphere, frozen areas like the polar ice caps, current and forecasted weather information, wind speed and direction, barometric pressure, and relative humidity. The main benefit to developers is getting access to hard to collect data that requires large infrastructure investments to gather and aggregate.

Topic 17: {location map maps information data code address locations service mapping}

This topic is describing mapping and location based information and services. Examples of API descriptions in this topic are Geobytes Get City Details and Map Data Services QuickMap API. Geobytes Get City Details API provides developers with the ability to query city attributes for a location given as n IP address or latitude/longitude coordinates. The main benefit for developers using this API is the ability to locate IP Addresses. Map Data Services QuickMap APIs on the other hand provide mapping functionality such as directions between locations and definition of lines and polygons within mapped regions. The main benefit to developers is reducing their development cost by providing them with functionality that is hard to develop in terms of know-how

Topic 18: {product products information online shopping deals search access price

store}

This topic describes shopping and product related information such as pricing and deals. An example API description for this topic is Macy's Catalog and Store Services API. The API allows developers to access different types of content provided by Macy's. Developers can get access to product catalogs, store events, promotions and user profiles. The main benefit provided by this API is allowing developers access to data that is costly to gather in an automated way. While developers can get access to most of this information by things like scraping and web crawling, the API standardizes the task and makes it easier for developers. Another example of an API in this topic is Macy's Shopping Bag Services. This API gives developers access to Macy's shopping bag and the services the shopping bag provides such as adding items, deleting items, or updating item quantities. The benefit this API gives to developers is the ability to build applications based on Macy's retail services. Applications will only allow the end users to access services provided by Macy's making the developers a part of Macy's service system acting as affiliates or resellers.

Topic 19: {voice call calls phone service applications services chat businesses integrate}

This topic describes the integration of voice calls and chat services into applications. Examples of API descriptions in this topic include Nexmo VoiceXML and Plivo call. These APIs give developers the ability to integrate functionality such as calling, text to speech conversion, conference calling, call tracking, and call recording. The main benefit to developers is the ability to integrate functionality that is hard to develop because of the

lack of know-how or the time needed for development.

Topic 20: {game games data information sports players live player statistics scores}

This topic describes access to game and sports statistics and information. Example API descriptions in this topic include Halo Profile API, Roanuz Cricket API, and SportRadar NHL API. The Halo Profile API gives developers access to player profile information for the game Halo. Both Roanuz Cricket API and SportRadar NHL API give developers access to real-time game stats and metrics for the respective sports intended to be used for betting applications. The main benefit to developers is getting access to hard to collect and aggregate sports data.

Topic 21: {data information open access government state public u.s tax states}

This topic focuses on open data and specifically government data. Some of the example API descriptions in this topic are CKAN API, It's Your Parliament EU Data, and Open State Project API. The main purpose of these APIs is to provide modeling and search capabilities to open data initiatives. The CKAN suite of API allows service providers to model their data and allow developers to search for the data. While It's Your Parliament EU Data API, and Open State Project API give developers the ability to query government data such as parliament records and state legislative information. The benefits of these types of APIs are twofold: they allow data providers to model and open their data in a standardized and consistent way, and they allow developers to access hard to collect and aggregate data.

Topic 22: {text language analysis content translation semantic sentiment machine words word}

This topic describes Natural Language Processing web services such as text analysis, translation, and sentiment analysis. Example API descriptions for this topic include the suite of Alchemy APIs, Idilia Language Graph API, and Proxem Ontology Based Topic Detection API. The purpose of these APIs is to give access to a range of Natural Language Processing functionality such as keyword extraction, predict language sentiment, and extract relevant topics from text based on relevant categories. The benefits these APIs provide to developers is the ability to do complicated NLP tasks without the need to develop the algorithms or invest in the infrastructure themselves.

Topic 24: {payment payments online card credit processing transactions account transaction cards}

This topic describes payment transactions processing. Examples of API descriptions in this topic include Allied Wallet API, Bitcoin Payment, and Openpay. These purpose of these APIs is to provide payment processing functionality for digital and real currencies. The benefit to developers using this API is getting the ability to process payments without having to deal with banking systems, compliance, fraud, and security.

Topic 26: {address data email addresses number service validation numbers phone verification}

This topic describes email address and phone number validation services. Example API descriptions in this topic include BankVal International API, Data8 Postcode Lookup API,

and FraudLabs MailBox Validator. These APIs allow developers to validate email addresses, validate routing numbers, and validate postcodes. The main benefits to developers is having access to functionality that is hard to implement due to the high cost of gathering the necessary data and the lack of know-how.

Topic 27: {video videos content photos media photo audio share sharing upload}

This topic describes content sharing and uploading services, and specifically video, photos, and audio. Examples of API descriptions in this topic include SlideShare Player API, Joomeo API, and Zenfolio API. These APIs allow developers to build applications that use functionality such as photo uploading, slide player, and photo editing. The main benefit to developers is the ability to integrate functionality that is hard to develop into their applications.

Topic 29: {data sequence database protein sequences analysis biological soap research genes}

This topic describes protein sequencing and genetic analysis functionality. API descriptions in this topic include Center for Biological Sequence Analysis API, ChromA API, and European Bioinformatics Institute API. The purpose of these APIs is to provide a programmatic interface to functionality such as protein sequencing, protein sequence analysis, alignment of mass spectrometry images for comparison, and DNA sequencing. The main benefit to developers is having access to very complex machine learning algorithms that are hard to develop due to lack of know-how and the high cost of needed infrastructure.

Topic 31: {data information traffic transit routes times public service bus time}

This topic describes access to transit routes and traffic information. API descriptions in this topic include TheBus API and Trafiklab suite of APIs. The purpose of these APIs is to provide access to real-time bus, subway, and train information such as location and schedule. These APIs also provide travel planning functionality. The benefit for developers using these APIs is having access to hard to gather information that only a city can provide.

Topic 32: {job learning students jobs information university student school system education}

Topic 32 describes access to information related to jobs, learning, and education. However, top examples of API descriptions for topic 32 are NAOqi DCM and Pemilu FAQ Presiden. Both are not related to the words in the topic. The first example is of a robotics API, while the second example is of open government elections data API for Indonesia. Other examples do include school information and data, as well as open data initiatives from school districts. Due to the inconsistency among examples this topic was excluded.

Topic 34: {project rest bot bots platform tools life projects goals management}

This topic describes functionality of project management tools and bots. Examples of APIs in this topic include the suite of Atlassian APIs and the suite of Gupshup APIs. Atlassian products provide a wide array of project management and collaboration functionality. Their APIs provide programmatic access to their functionality. While

Gupshup suite of APIs provide access to Gupshup bots for chatting and sms. Both providers focus on workplace tools and applications such as project management, collaboration and chatting. The main purpose of the APIs is to enable the users of the products to customize, and integrate the products with other applications. The main benefit these APIs provide is interoperability and personalization of the main service provided by the vendors.

Topic 35: {images image files file pdf documents upload recognition document conversion}

This topic describes file conversion functionality. The top example of an API description for this topic is ConvertAPI. ConvertAPI allows developers to perform file format conversions between multiple file formats. The main benefits to the developers is access to hard to develop functionality in terms of know-how and time required.

Topic 36: {bitcoin exchange trading service account trade currency calls orders information}

This topic describes cryptocurrency information and functionality. Example API descriptions in this topic include 50BTC API, BitMarket API, and Bitcurex API. These APIs allow developers to perform cryptocurrency related tasks such as mining bitcoin, query exchange rate information and market statistics, and perform trading transactions like purchasing, holding, and selling bitcoins. The main benefit to developers is the ability to have access to hard to gather data and hard to develop functionality.

Topic 39: {sms messages send messaging text mobile message bulk sending service}

This topic describes sms messaging functionality. Examples of API descriptions in this topic include 24X7SMS, 5Star SMS, and BulkSMSVilla. These APIs provide developers the ability to send single or bulk sms messages to multiple countries and mobile providers around the world. The main benefit to developers is having connectivity to multiple mobile network operators at once, reducing the cost of connectivity and integration.