Measuring the impact that domain knowledge in the form of simple ontology structures has on predictive modelling processes in the context of academic library virtual reference services.

By

Jeremy Walker

Thesis Project
Submitted in partial fulfillment of the
Requirements for the degree of

MASTER OF SCIENCE IN DATA SCIENCE

June, 2019

Alianna J. Maren, First Reader

Jason Coleman, Second Reader

ABSTRACT

Measuring the impact that domain knowledge in the form of simple ontology structures has on predictive modelling processes in the context of academic library virtual reference services.

Jeremy Walker

Using transcripts from Kansas State University Libraries' (KSUL) virtual reference services (VRS), an experimental design was created to investigate the value and impact of different natural language processing techniques in the context of creating predictive models. Models were created to use only the first few word tokens supplied by VRS patrons and predict if the overall VRS interaction would be labelled as "easy" or "hard" by VRS operators. The experimental design incorporated machine learning methods (LDA and Doc2Vec), rules-based text processing (TF-IDF), and ontology structures as parameters in the modelling processes. With a specific focus on ontology structures, experimental results indicate that incorporating domain knowledge into predictive modelling processes contributes in significant and positive ways to overall model performance. Results also demonstrated that machine learning processes like Doc2Vec are capable of capturing meaningful representations of domain knowledge in abstract quantified vectors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Project Context

At Kansas State University Libraries (KSUL), a team of librarians, staff, and student employees provide library patrons and visitors with a wide variety of services and support referred to collectively as "Ask-A-Librarian". One component of these services is a virtual reference service (VRS) in which library patrons use an instant messenger platform to chat synchronously with library staff. As a component of the VRS operations, KSUL maintains a database of almost every VRS transcript and accompanying metadata labels supplied by the VRS operator. For all service interactions, virtual and otherwise, library staff record basic information about the service interaction and assign a variety of classification labels and a 'difficulty' rating. In the case of chat and email services, the operator copies the full text of the entire interaction into the database (see Figure 1).

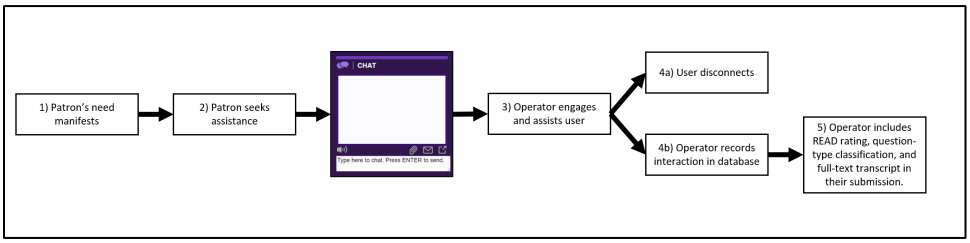

Figure 1: VRS Operations Flowchart

With respect to VRS at KSUL, the database of transcripts and associated metadata constitutes a robust and rich dataset that can be leveraged to enhance future library services. Through the application of natural language processing techniques, it should be possible to develop a predictive model that uses only the initial few words or text strings provided by VRS patrons to reliably predict the final 'difficulty' rating, or an approximation thereof, before the VRS interaction fully commences. Hypothetically, such a

model could then be incorporated into KSUL's VRS operations by interceding every incoming VRS patron's inquiry, making a prediction, and then routing the patron to an appropriate VRS operator (see Figure 2).

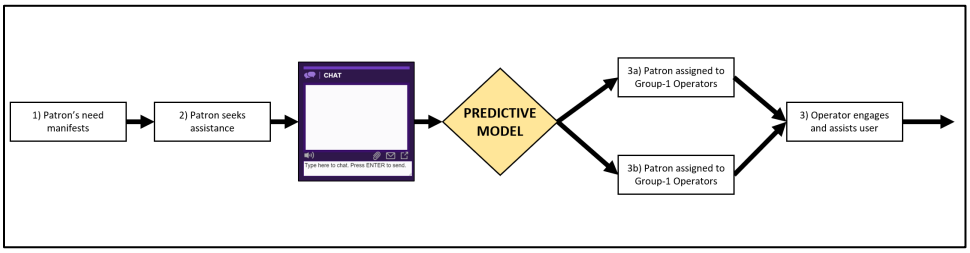

Figure 2: VRS Operation Flowchart, Including Predictive Model

Research Objective

Although the full implementation of the predictive/prescriptive modelling process described above would necessarily entail a myriad of strategic business decisions and engineering challenges, those issues are not the focus of this research project. Instead, the primary focus is centered on the evaluation of modelling processes and parameters used to process raw VRS transcripts, represent the data in meaningful ways using both rules-based methods and machine learning algorithms, and using said representation to generate predictions.

Specifically, a key modelling process that will be investigated is the incorporation of domain knowledge, in the form of simple ontological structures, into the modelling process as a form of manual feature engineering. Although automated machine learning processes are at the cutting edge of natural language processing, many of these tools require datasets that are immensely larger than the KSUL dataset. Additionally, as noted by Craig Boman in a report commissioned by the American Library Association, "…few libraries and fewer librarians are prepared to take full advantage of the benefits of using AI" (Boman, 2019, p.21). In this context, it may prove critical to enable librarians, who may not be

experts in the latest trends in machine learning and artificial intelligence, to incorporate their domain knowledge directly into the modelling process.

Therefore, although this project will evaluate the value and impact of a variety of modelling processes and parameters, the primary goal of this project is to determine the extent to which the incorporation of domain knowledge, as reflected in the structure and use of a simple ontology, into the modelling and representation of VRS transcript data has a recognizable and significant impact on predictive modelling processes. Or, stated formally in terms of null and alternative hypotheses:

> *H_0 : The impact of incorporating ontology labels on the performance metric of a predictive modelling process is not significantly different from zero.*
>
> *H_1 : The impact of incorporating ontology labels on the performance metric of a predictive modelling process is significantly different from zero.*

Brief Dataset Overview

The original KSUL dataset provided by the services manager at KSUL contains 15,690 VRS records. However, due to inconsistencies with how VRS operators record and submit data, and due to technical malfunctions in the VRS system that occasionally result in loss of data, not all records are fully useable for this project. Some records are duplicates, some records do not contain original VRS transcripts, and some records have data that are so extensively edited, redacted, or modified as to render the record unusable. Following extensive data cleaning and preparation, 14,604 records with transcripts remained and were useable for the first stages of analysis and modelling. The VRS transcripts contained in the KSUL dataset cover a span of five years, ranging from 2013 to 2018. For details regarding the pre-processing and exploratory analysis of the KSUL dataset, see section: DATA PREPARATION & EXPLORATION.

User Privacy, Data Availability & IRB Approval

A critical administrative note with respect to this project concerns VRS patrons' rights to privacy and expectations of anonymity. By default, the KSUL VRS system does not collect personally identifying information on patrons. The data preparation stages of this project were designed to ensure that any incidental personally identifying information provided by the patron was fully redacted from both the raw data and representations built upon said data. However, because it is not possible to fully guarantee that 100% of identifiable VRS data has been expunged from the raw data, the data and models for this project will not be shared openly.

Institutional Review Board (IRB) review was sought at Northwestern University. The university's IRB office determined this research project to not be classified as "Human Research" and did not require further or ongoing review. The Northwestern University IRB-ID for this project is STU00207931.

LITERATURE REVIEW

VRS TRANSCRIPT ANALYSES

Qualitative Coding & Operator Behaviors

The vast majority of research conducted by libraries and librarians on VRS transcripts has been inferential and qualitative in nature. In studies by Ward (2003); Burger et al. (2009); Valentine and Moss (2017), the researchers evaluated operator behavior and communication style with respect to established service guidelines like RUSA's (Reference and User Services Association) "Guidelines for Behavioral Performance of Reference and Information Service Providers" (RUSA, 2008b). In other studies, such as those by Radford (2006); Meert and Given (2009); Koshik and Okazawa (2012); and Baumgart et al. (2016), researchers developed their own standards and coding schema for coding and evaluating operator behavior and communication.

In each of the studies above, librarians were directly responsible for labelling and evaluating transcripts with no external validation from patrons. Although this is a common trend in the research pertaining to virtual reference services, other studies have included analysis and data gathered from external parties. In both Waugh (2013) and Jacoby et al. (2016), researchers incorporated patron feedback and assessment into their evaluation of chat transcripts. However, like with the other studies, the researchers were primarily focused on the relationship between VRS quality and the choices operators made as they interacted with library patrons.

Based on the preponderance of literature surrounding the coding and evaluation of virtual reference transcripts based on behaviors, chat transcripts represent a potentially rich source of data for creating predictive models that can identify and encode operator behaviors. There is evidence that this application of natural language processing already exists in the context of mental health counseling and

support (Althoff, Clark, & Leskovec, 2016). Unfortunately, this approach is not possible with the KSUL dataset in its current form since it lacks sufficient qualitative and evaluative labels for the purposes of building models around operator and patron behaviors.

Qualitative Coding & User Behavior

Other research into VRS has focused on the core concepts and topics that manifest from patron behavior throughout VRS interactions. For example, in one study, researchers identified broad categories of question-types that patrons brought to VRS interactions (Morais & Sampson, 2010). In another example, librarian researchers analyzed chat transcripts for the purposes of evaluating library website usability and patrons' perceptions of library resources (Powers, Shedd, & Hill, 2011). In both studies, analyses relied heavily on subjective, qualitative coding.

Although somewhat dated now, further and significantly more robust evidence of the qualitative and inferential nature of VRS research is provided by a systematic review from 2011 in which the authors surveyed 59 documents containing an aggregate of 149 research questions (Matteson, Salamon, & Brewster, 2011).

Distinguishing Among Categories of Chat Operators

Another vein of research pertaining to reference services, and VRS in particular, is the evaluation and analysis of discrete categories of VRS operators. "Librarians", "Paraprofessionals", "Assistants", "Student Assistants" and many appropriate synonyms for the previous terms are used to categorize different types of VRS operators with respect to varying notions of cost, skill, and function within a library's operations. Definitions and distinctions are different in different studies and different institutions, but a common theme in the literature is that these differences present both challenges and opportunities for managing VRS.

One common topic that emerges in the literature is issues surrounding VRS operators' referral of patrons to other operators. Bracke et al. (2007); Fuller and Dryden (2015); and Maloney and Kemp (2015) all describe challenges associated with VRS operators consistently, competently, and quickly referring patrons to operators. These referrals are often framed in terms of patrons being forwarded from less specialized to more specialized operators, or from non-Librarians to librarians. King and Christensen-Lee (2014) discuss these issues in the context of distinguishing between "full-time" and "part-time" reference librarians.

The starkest research concerning different types of VRS operators pertains to the use of undergraduate student employees as VRS operators in academic libraries. In a study by Bravender et al. (2011), the researchers argue that the majority of analyzed VRS interactions do not require fully skilled librarians and that student operators are capable of fulfilling service expectations. They emphasize that use of students is more cost effective than use of librarians. In research conducted by Lux and Rich (2016) conclude that although student employees can provide excellent VRS service if given proper training, they are notably deficient in referring patrons to more specialized or appropriate operators. This is further confirmed by research conducted by Keyes and Dworak (2017) in which the authors conclude that although students are able to provide high quality VRS service, there is a need for ongoing training regarding referring patrons to other operators. One critical implication of Lux and Rich (2016) and Keyes and Dworak's (2017) research is that there may be immense value to be gained from the development and implementation of predictive models which can automatically refer and route incoming VRS inquiries to appropriate operators.

ARTIFICIAL INTELLIGENCE IN LIBRARIES AND VRS

Machine Learning Use in Libraries and VRS

As is the case with many other industries and professions, artificial intelligence and machine learning are topics that are increasingly salient in the context of library services and operations. One central piece of evidence comes from a 2019 report from the American Library Association titled *Artificial Intelligence and Machine Learning in Libraries* (Boman, Kim, Yelton, & Griffey, 2019). Throughout the report, section authors reported on projects, challenges, and movements within the library profession. Subjects included topic-modelling using LDA, generating document embedding using Doc2Vec to enhance information discovery platforms, ethics and privacy concerns underpinning modelling processes, and the creation of library-managed AI labs on university campuses.

Further evidence of growing library interest in machine learning is represented by the relatively new *Projects in Artificial Intelligence Registry* (PAIR) online repository hosted by the University of Oklahoma (University of Oklahoma Libraries, 2019a). The PAIR repository, designed to be a community forum for sharing information about library-centric machine learning projects, does not have many projects listed. However, two of the projects listed on the website are about the implementation of "chatbots" in the context of library VRS operations. On the PAIR website, the two separate chatbot projects, one developed at the University of Oklahoma Libraries (OU-L) and the other at the University of California, Irvine Libraries (UCI-L), are described as being intended to automatically respond to and answer common, simply, and otherwise low-level patron questions in a VRS environment (University of Oklahoma Libraries, 2019b, 2019c).

Unfortunately, there appears to be little concrete modelling information provided about the underlying machine learning techniques and models employed in either of these projects. For the OU-L

project, other than a public test interface, there is no publicly available code or project analysis available (University of Oklahoma Libraries, 2019c). For the UCI-L project, the researchers' chatbot was implemented using an automated live-chat tool called Program-O (Kane, 2019). Although the documentation associated with Program-O is sparse and it is unclear from the documentation what, if any, artificial intelligence techniques are used to drive conversation, the core design of Program-O and the UCI-L's implementation of the chatbot revolves around mapping common sets of questions to canned responses using Artificial Markup Language files (Kane, 2019; Program O, 2019). This very closely resembles the structure of a shallow ontology. The generally positive results reported in Kane's (2019) research regarding VRS service quality and characteristics indicate that there is value in incorporating domain knowledge into machine learning modelling processes involving VRS.

Machine Learning Analysis of VRS

To date, one of the only instances in which machine learning algorithms have been used in the context of VRS comes from the *17th Annual Brick & Click Libraries Conference Proceeding*. In a conference presentation, Kohler (2017) shared a variety of use-cases for different natural language processing methods and techniques, and described how they could pertain to VRS. Specifically, Kohler's demonstration highlighted the use of various topic modelling techniques for topic extraction and the use of standalone sentiment libraries for evaluating sentiment within VRS transcripts. Most importantly, the presentation included data and findings showing that the application of topic modelling algorithms could identify latent features that correlated with difficulty ratings (see 'READ Scale' in next section) provided by VRS operators (Kohler, 2017). Although the information available from the conference proceedings is relatively shallow, this research indicates that the KSUL dataset may be well suited to developing predictive models built upon machine-learned data representations.

LIBRARY REFERENCE SERVICES METADATA & VARIABLES

READ Scale

Among the most intuitively valuable metadata available in the KSUL dataset are the READ Scale ratings that are assigned to most of the recorded transcripts. The READ (Reference Effort Assessment Data) Scale was first introduced in 2007 by Gerlich and Berard and piloted at Carnegie Mellon University. The scale is a qualitative 6-point ordinal scale that was originally developed to supplement shallow quantitative measures of reference services by doing more than simply counting the number of service interactions at a library service point. The goal of the READ Scale was to provide a better way to capture and record the skill, expertise, effort, and time expended by reference librarians for every individual interaction with library patrons.

In a follow-up study, the same authors conducted a nation-wide (US) study of the viability and utility of using the READ Scale at a variety of different institutions. Although the authors note that responses from librarians at testing institutions indicated a broad acceptance and interest in continuing to use the READ Scale, the direct feedback shared by the authors indicates that there are methodological challenges built into the scale. For instance, the authors report that multiple respondents, both in survey and free-form responses, expressed that they struggled with understanding and normalizing the subjective thresholds that distinguish between different points on the scale (Gerlich & Berard, 2010).

Overall, based on the research reports by Gerlich and Berard's (2007; 2010) the READ ratings present in the KSUL dataset may be used as an effective proxy of "difficulty." However, the validity and reliability of the measure is not perfect due to the subjectivity associated with each librarian's qualitative assessment of individual reference interactions.

Question Types

The second key set of metadata in the KSUL dataset that may be valuable to developing models is the Question Type field. Although the KSUL dataset includes twelve distinct classifications within the Question Type field, those distinctions are largely localized to the needs of KSUL. National organizations such as RUSA and ARL (Association of Research Libraries) define library service operations much more broadly into "reference" and "non-reference" classifications.

According to RUSA, a division of the American Library Association, "reference transactions" are defined so as to exclude library service interactions that revolve around directional, circulation, policy, or technical library services and operations (RUSA, 2008a). Furthermore, the most recent ARL survey provides instructions that define "reference transactions" as "an information contact that involves the knowledge, use, recommendations, interpretation, or instruction in the use [or creation of] one or more information sources by a member of the library staff." The ARL definition explicitly excludes directional questions from being counted as "reference transactions" (ARL, 2019). Although not explicitly defined, it can be inferred that the authors of the ARL document would also exclude technical questions pertaining to printers, scanners, and other equipment from inclusion in the "reference transaction" classification.

In both instances, the RUSA and ARL binary definitions of "reference" and "non-reference" question classifications indicate that librarians perceive an intrinsic and valuable distinction between these two broad categories of patron interactions. Although KSUL's dataset includes significantly more granular classifications, the same trend is present (see Table 1). For example, the "Reference" and "Research Consultation" question types in the original KSUL dataset are the only two categories that clearly connect to the ARL and RUSA definitions of "reference." The remaining question types in the KSUL dataset either clearly represent "non-reference" questions or are not clearly connected to either of the two broad categories. For further details and definitions pertaining to question classification within the KSUL dataset, see DATA PREPARATION & EXPLORATION and METHODOLOGY sections.

QUESTION TYPES: REFERENCE VS. NON-REFERENCE	
Original KSUL Question Types	*Question Type Grouping*
BUILDING	NON-REFERENCE
CIRCULATION	NON-REFERENCE
COPYRIGHT	NON-REFERENCE
DIRECTIONAL	NON-REFERENCE
K-Rex	UNCLEAR
KAPI	UNCLEAR
MISC	NON-REFERENCE
NEW PRAIRIE PRESS	UNCLEAR
REFERENCE	REFERENCE
RESEARCH CONSULTATION	REFERENCE
RESERVES	NON-REFERENCE
TECHNICAL	NON-REFERENCE

Table 1: Question Types - Reference vs. Non-Reference

TEXT PROCESSING AND UNSUPERVISED MACHINE LEARNING METHODS

With any data analytics process, especially in natural language processing tasks, some of the first key steps revolve around how to transform, filter, and represent the raw data in a meaningful way for the purposes of prediction and classification.

Common Text Processing Techniques

There are a variety of common techniques for filtering and extracting relevant terms from raw text. Weiss et al.'s text *Fundamentals of predictive text mining* (2015) provides robust overviews of these methods. The most critical, and simple, methods involve eliminating words that are too frequent or too rare within a corpus. Weiss et al. (2015) also argue that retaining only a small subset of several hundred of the most prominent terms is sufficient for most tasks. Additionally, the authors note that in place of using raw frequency counts, weighted metrics like TF-IDF ("term frequency – inverse document

frequency") are valuable for conveying a sense of individual terms' relative importance within a corpus. They contend that although multiple scaling formulations exist, the general principle leads to greater prominence being allocated to terms that are more discrepant and important within a corpus than otherwise highly frequent but irrelevant terms.

The strongest sign of the value of these approaches is reflected in the fact that prominent text mining and natural language processing programming packages support these functions explicitly. In python packages such as Scikit-Learn and Gensim, operations like finding term frequencies, reweighting terms, truncating term dictionaries, and eliminating predefined and custom "stop word" lists are a rote component of preprocessing functions (Pedregosa et al., 2011; Řehůřek & Sojka, 2010).

Latent Dirichlet Allocation

First introduced by Blei et al. (2003), latent dirichlet allocation (LDA) is a probabilistic model that generates topic-document and word-topic distributions given a fixed number of assumed latent topics. The authors argue that LDA, like latent semantic indexing (LSI), is effective as a dimension reduction technique, but better suited than LSI for representing the underlying semantics of natural language data (Blei, Ng, & Jordan, 2003). The immense popularity of LDA as a topic modelling technique is evidenced by a recent survey conducted by Jelodar et al. (2018) in which the authors review over two hundred articles concerning different implementations and formulations of the original LDA model.

Although LDA is primarily known for its power as an unsupervised latent feature modelling technique, the algorithm's value as a dimensionality reduction technique also makes it useful for supervised learning models. As noted by Chang et al. (2009), LDA-generated topic distributions that are not inherently interpretable by humans may serve as valuable representations of data for the purposes of prediction. Use of LDA representations of data for training supervised models has been shown to be

effective in relationship to news, social media profiles, and criminal analysis (Al-Salemi, Ab Aziz, & Noah, 2015; Liu, Wang, & Jiang, 2016; Wang, Gerber, & Brown, 2012). And, directly related to KSUL's dataset, Momtazi (2018) provides a recent example of LDA being used to enhance classification models in the context of online question-answer forums.

Word2Vec and Doc2Vec

Two other methods for generating unsupervised representations of natural language data are the Word2Vec and Doc2Vec neural network architectures. Word2Vec, developed by researchers at Google, was designed as an efficient method for learning fixed-length vector representations of individual words that retained meaningful positioning relative to other semantically related words (Mikolov, Chen, Corrado, & Dean, 2013). Expanding on this research, Doc2Vec was developed as a neural network architecture for learning fixed-length vector representations of sentences, paragraphs, and documents of variable length (Le & Mikolov, 2014).

As with LDA, the Word2Vec and Doc2Vec algorithms have shown promising results as unsupervised learning techniques for creating word and document representations ('embeddings') that can enhance and improve performance in predictive and classification tasks. Recent examples include applications in clinical narratives, genetics research, and news analysis (Lauren, Qu, Zhang, & Lendasse, 2018; Oubounyt, Louadi, Tayara, & Chong, 2018; Sinoara, Camacho-Collados, Rossi, Navigli, & Rezende, 2019).

Domain Knowledge & Ontologies

Beyond purely rules-based and algorithmic approaches to transforming and representing language data, there is extensive and commonly understood value in incorporating concrete domain knowledge into natural language processing tasks. For instance, multiple approaches for incorporating

metadata tags and labels into LDA models have been proposed (Ramage, Hall, Nallapati, & Manning, 2009; Zhu, Blei, & Lafferty, 2006). Furthermore, researchers have developed algorithms and approaches to generating ontologies, pseudo-ontologies, and otherwise structured graph-representations of underlying textual data (Ibrahim & Ahmad, 2010; Kozareva, 2014; Tanev, 2014). Additionally, there are multiple methods available for visualizing ontological structures (Katifori, Halatsis, Lepouras, Vassilakis, & Giannopoulou, 2007).

Although these and other advanced methods are available, any predictive model built upon the KSUL dataset is better suited to simpler and more foundational ontological structures and assessment. A technical report from Stanford University, *Ontology Development 101: A Guide to Creating Your First Ontology*, represents a starting point for amateurs (in this case, librarians) seeking to create ontologies (Noy & McGuinness, 2001). The report includes thorough explanations of how to understand the core components of ontological structures and how to approach creating ontologies from scratch. When combined with qualitative coding efforts already present in research concerning VRS, the formal definition and incorporation of ontologies may aide in the development of predictive models involving VRS transcripts. Evidence of the utility of ontologies in predictive modelling is found in customer service research (Iwashita, Shimogawa, & Nishimatsu, 2011).

The evaluation of ontological structures and reasoning in the modelling process is critical to understanding if ontological concepts are being captured or represented in the data. One research paper in particular provides grounded methods for evaluating the presence, strength, and representation of ontological nodes and domain-knowledge within text and document embeddings (Alshargi, Shekarpour, Soru, Sheth, & Quasthoff, 2018). Specifically, the authors' discussion of a 'Categorization metric' as a method for evaluating document embeddings with respect to ontology concepts should be extremely useful for evaluating the overall quality of VRS data representations (Alshargi et al., 2018).

DATA PREPARATION & EXPLORATION

TRANSFORMING AND PREPARING RAW DATA

Data Protection & Shared Examples

All examples of raw or transformed data drawn from the KSUL dataset presented in this report are both A) individual representations of processes and analyses used in this research and B) presented based on the condition that they do not contain any form of personally identifying information. Further examples from the whole dataset are unavailable to general audiences due to the small, but real, risk that personally identifiable information is still present in the data.

Data Source Wrangling and Parsing

All of the data in the KSUL dataset was extracted from KSUL's instance of a reference services tracking database; LibAnalytics. As part of KSUL's public service operations, all library employees are expected to record a variety of metadata and descriptive information about every single interaction they have with library patrons that is not otherwise recorded by the library's circulation software. In the context of VRS, KSUL's operators are expected to copy and paste VRS transcripts in their entirety into LibAnalytics along with accompanying metadata.

Consequently, one challenge posed by how the KSUL dataset was developed is that there is a significant amount of operator error manifest in the dataset. Specifically, because VRS operators must manually copy and paste transcripts into LibAnalytics, and not all operators record data in the exact same way, it was necessary to spend a significant amount of time wrangling the data into a consolidated format. Once the data was sufficiently organized, raw transcript data would appear as a single large string (see Figure 3). After a variety of parsing operations, the transcript was formatted into programmatic lists of

lists representing the timestamp, patron or operator label, and the accompanying language data (see Figure 4). All of the patron's supplied text was then assembled into a single sequential string which would form the base input for predictive modelling (see Figure 5).

9:00 64768850013983911682632 62 I'm looking for a 19th century article about women fashion in France but cant seem to find anything? Do you have any good links? 9:00 me hello 9:01 me Hmm...I have a few ideas! 9:01 me Are you looking for articles that are about 19th centruy french fashion OR articles written in the 19th century about french fashion? (the former will definitely be easier I think) 9:02 64768850013983911682632 62 I am looking for scholarly articles about 19th century French fashon 9:02 64768850013983911682632 62 Mainly women 9:04 me Ok, let me see what I can find! 9:04 64768850013983911682632 62 Thanks you rock! 9:04 me I am going to start with our Search It tool. Also, we have some fashion databases as well http://apps.lib.k-state.edu/databases/category/human-ecology/apparel-textiles/ 9:04 me Have you had a chance to try either of those sources? 9:05 64768850013983911682632 62 Yeah I have tried.. I am no sure if im too specific or not specific enough. 9:06 me Gotcha. Also, do they have to be scholarly articles? Would library books work as well? 9:06 64768850013983911682632 62 Yes, I believe so. 9:07 me I found one promising book in Search It "Accessories to modernity : fashion and the feminine in nineteenth-century France" 9:07 64768850013983911682632 62 My assignment details just say two scholarly sources. 9:09 me Ok, I certianly think many of the books in the library qualify as "scholarly" Obviously some will not (ex. Batman comics), but I think you should be able to identify if a book is a scholarly source (they will have lots of references, detailed info, neutral tone, etc...) 9:10 me Here is a quick video showing how I found some books.... 9:10 me http://screencast.com/t/s8gDSF1fm5C 9:12 64768850013983911682632 62 Okay, Thank you so much! 9:12 me In that video I highlighted the call number for the book 9:12 me Call numbers are ordered by subject, so if you can find that book, you should be able to find many other relevant books right next to it 9:12 me Also, for research articles, I think the "Berg Fashion Library" databases may be another good place to search 9:13 64768850013983911682632 62 Thank you, I appreciate it. 9:14 me Does that give you a good starting point? 9:16 64768850013983911682632 62 Yes, Thanks! 9:17 me Great! Please don't hesitate to come back if you have more questions 9:26 64768850013983911682632 62 Awesome thank you!

Figure 3: Raw VRS Transcript

['9:00', 'patron', "I'm looking for a 19th century article about women fashion in France but cant seem to find anything? Do you have any good links?"]

['9:00', 'staff', 'hello']

['9:01', 'staff', 'Hmm...I have a few ideas!']

['9:01', 'staff', 'Are you looking for articles that are about 19th centruy french fashion OR articles written in the 19th century about french fashion? (the former will definitely be easier I think)']

['9:02', 'patron', 'I am looking for scholarly articles about 19th century French fashon']

['9:02', 'patron', 'Mainly women']

['9:04', 'staff', 'Ok, let me see what I can find!']

['9:04', 'patron', 'Thanks you rock!']

['9:04', 'staff', 'I am going to start with our Search It tool. Also, we have some fashion databases as well http://apps.lib.k-state.edu/databases/category/human-ecology/apparel-textiles/']

['9:04', 'staff', 'Have you had a chance to try either of those sources?']

['9:05', 'patron', 'Yeah I have tried.. I am no sure if im too specific or not specific enough.']

['9:06', 'staff', 'Gotcha. Also, do they have to be scholarly articles? Would library books work as well?']

['9:06', 'patron', 'Yes, I believe so.']

['9:07', 'staff', 'I found one promising book in Search It "Accessories to modernity : fashion and the feminine in nineteenth-century France"']

['9:07', 'patron', 'My assignment details just say two scholarly sources.']

['9:09', 'staff', 'Ok, I certianly think many of the books in the library qualify as "scholarly" Obviously some will not (ex. Batman comics), but I think you should be able to identify if a book is a scholarly source (they will have lots of references, detailed info, neutral tone, etc...)']

['9:10', 'staff', 'Here is a quick video showing how I found some books....']

['9:10', 'staff', 'http://screencast.com/t/s8gDSF1fm5C']

['9:12', 'patron', 'Okay, Thank you so much!']

['9:12', 'staff', 'In that video I highlighted the call number for the book']

['9:12', 'staff', 'Call numbers are ordered by subject, so if you can find that book, you should be able to find many other relevant books right next to it']

['9:12', 'staff', 'Also, for research articles, I think the "Berg Fashion Library" databases may be another good place to search']

['9:13', 'patron', 'Thank you, I appreciate it.']

['9:14', 'staff', 'Does that give you a good starting point?']

['9:16', 'patron', 'Yes, Thanks!']

['9:17', 'staff', "Great! Please don't hesitate to come back if you have more questions"]

['9:26', 'patron', 'Awesome thank you!']

Figure 4: Parsed VRS Transcript

I'm looking for a 19th century article about women fashion in France but cant seem to find anything? Do you have any good links? I am looking for scholarly articles about 19th century French fashon Mainly women Thanks you rock! Yeah I have tried.. I am no sure if im too specific or not specific enough. Yes, I believe so. My assignment details just say two scholarly sources. Okay, Thank you so much! Thank you, I appreciate it. Yes, Thanks! Awesome thank you!

Figure 5: Full patron-supplied text

Data Redaction

As a core ethical component of research involving KSUL's dataset, efforts were made to redact any personally identifiable information remaining in the VRS transcripts. For example, for all instances in which a patron or operator stated, "my name is ____", the subsequent space-delimited string of text was replaced with "nameredacted". The same policy resulted in email addresses being replaced with "emailredacted". In many instances, operators used common strings like "xxx" and "***" to redact personally identifying information while they were recording the VRS transcript in LibAnalytics. These instances were also replaced with a unifying string: "redactedinfo".

Additionally, system-generated noise was fully redacted from the transcripts and not replaced with any strings or tokens. For example, the VRS system used at KSUL prints a variety of messages directly into the chat interface and these strings are often included in the operators' submissions into LibAnalytics. Examples of system-generated text include "…has left the conversation", "transfer from…", and "answered by…". When these statements are system-generated and properly detected within the transcript, they carry zero value for any form of analysis. In order to reduce noise present in the data, these strings were removed entirely.

Sample Removal

After significant effort spent preparing and organizing the KSUL dataset, further inspections of the data revealed that 1,083 samples needed to be removed from the study. Multiple problems manifest that led to samples not being included in the final dataset:

- Duplicate samples/records were removed.
- Samples in which the full transcript was not available were removed.
- Samples that did not clearly distinguish between raw transcripts and operator annotations were removed.
- Samples that were so heavily edited that they could not be parsed were removed.
- Samples that reflected operators using the VRS system to communicate internally were removed.

Some of the samples that were removed from the dataset were identified manually upon close inspection of the raw data. Other samples were identified programmatically.

All of the processes and scripting used to organize and modify individual samples in the KSUL dataset are documented in APPENDIX I – DATA PREPARATION.

HIGH-LEVEL DATA EXPLORATION

READ Scale Rating

At the end of every VRS interaction with patrons, KSUL's VRS operators are expected to label each interaction with a READ score based on the scale developed by Gerlich & Berard (2007). The scale is a qualitative ordinal scale ranging from "1" representing the easiest questions to "6" representing the most difficult questions. From the dataset, 2,690 (18%) are not labelled with a READ scale rating. Although unlabeled VRS transcripts could not be used for validating any supervised learning, the unlabeled VRS transcripts themselves could still contribute to unsupervised learning processes that would help develop

representations of the data. The following table and figure provides a high-level snapshot of the distribution of READ Scale ratings present in the KSUL dataset (see Table 1 and Figure 6).

READ Scale Ratings	
Rating	***Count***
1	3355
2	5431
3	2532
4	493
5	89
6	17
Unknown	2690
TOTAL	14607

Table 2: READ Scale Rating by Count of VRS Transcripts

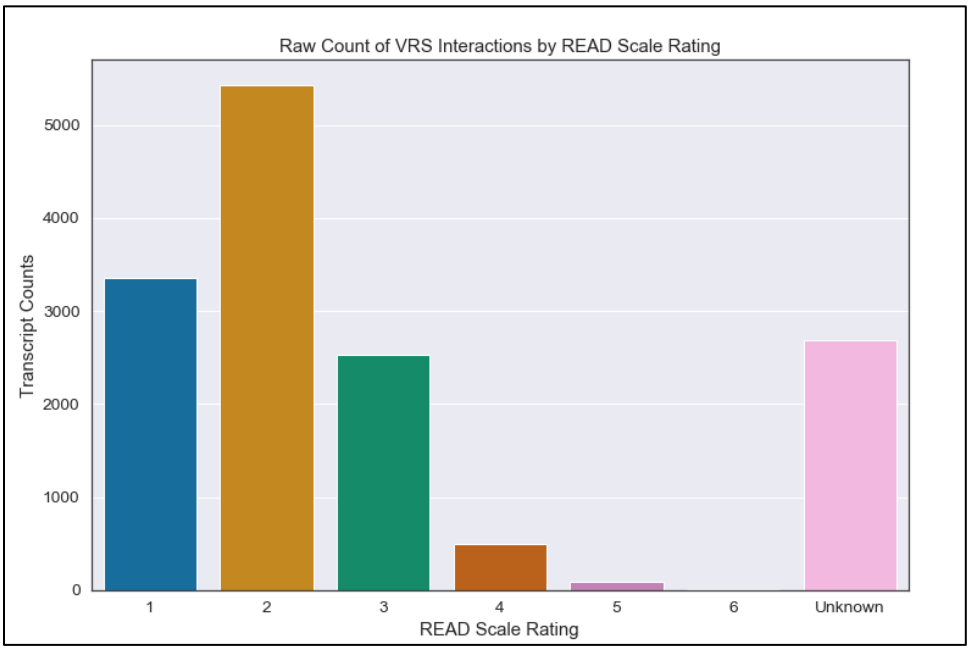

Figure 6: READ Scale Rating by Count – Plot

Question Types

The other primary label that KSUL's VRS operators are expected to provide for each interaction is the "Question Type" category. KSUL's service operators are expected to follow the definitions established by service managers when recording patron interactions in LibAnalytics (see Table 2). Cursory review of VRS transcript counts and labels shows an extremely uneven distribution of "Question Type" classifications. For instance, the "Reference" label represents approximately 66% of the entire KSUL dataset whereas niche classifications like "KAPI" and "NewPrairiePress" each represent a small fraction of a percent of the total dataset (see Table 3 and Figure 7).

QUESTION TYPES: DEFINTIONS	
BUILDING	Issues reported to us about the building and objects inside it. Examples: Fire alarms. Toilets/urinals. Noisy people.
CIRCULATION	Questions about circulation policies or processes.
COPYRIGHT	Questions about copyright.
DIRECTIONAL	A purely directional question or help figuring out where something is located in the library (e.g., where is BF425.F3).
K-Rex	Questions about K-Rex.
KAPI	Questions about the Kansas Aerial Photography Initiative (see http://www.lib.k-state.edu/apps/kapi/kapi.php)
MISC	Questions about supplies or general policies. When in doubt, choose misc.
NEW PRAIRIE PRESS	Questions about New Prairie Press.
REFERENCE	A question about how to find information on a topic or how to find a known information source or how to cite sources.
RESEARCH CONSULTATION	An involved transaction in which your goal is to help the patron learn several strategies and/or resources for finding or using information now and in the near future.
RESERVES	Questions about reserves policies or processes.
TECHNICAL	Questions about equipment or software or hardware (Examples: help with printers, scanners, copiers).

Table 3: Question Type Defintions

Question Type	
Type	***Count***
Reference	9773
Technical	1318
Misc	1246
Building	959
Circulation	659
Directional	308
Reserves	155
ResearchConsultation	80
Unknown	63
KREx	16
Copyright	14
NewPrairiePress	9
KAPI	7
TOTAL	14607

Table 4: Question Type by Count

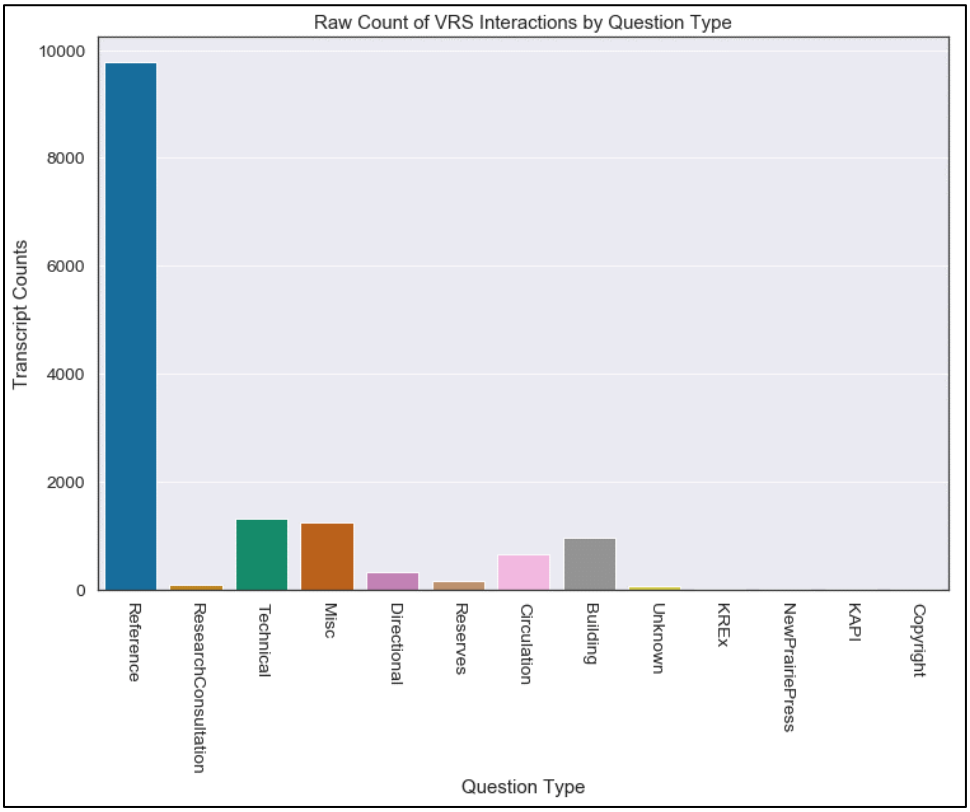

Figure 7: Question Type by Count – Plot

READ Scale Ratings x Question Types

Although the distributions presented by the aggregate READ Scale ratings and Question Type classifications are heavily skewed and uneven, respectively, cross tabulation of these two key pieces of metadata from the KSUL dataset reveals that the data may lend itself well to predictive modelling applications. Inspecting the distribution of READ Scale ratings within each Question Type category, slight patterns being to emerge (see Table 4, Table 5, and Figure 8). "Reference" questions are the only category with sizeable proportions of READ Scale ratings above 2. In contrast, "Building", "Directional", and "Misc" questions are markedly skewed in favor of READ Scale ratings of 1.

These patterns echo the results identified by Kohler's (2017). Her research noted that once VRS transcripts had been decomposed into latent-feature sets, the latent features ("topics") closely tracked with different READ Scale ratings (Kohler, 2017). Even though the Question Type and READ Scale ratings both represent post-hoc qualitative labels, patterns that suggest different types of VRS interactions are correlated with different READ Scale ratings suggest that the underlying VRS transcripts may contain data signals that can be quantified, represented, and predicted in reliable ways.

READ Scale Ratings vs. Question Type: Transcript Counts									
		READ							
		1	2	3	4	5	6	Unknown	TOTAL
Question Type	Building	507	251	19	2	0	0	180	959
	Circulation	208	286	60	6	1	0	98	659
	Copyright	0	8	1	1	0	0	4	14
	Directional	151	74	10	0	0	0	73	308
	KAPI	1	3	2	0	0	0	1	7
	KREx	4	8	2	0	0	0	2	16
	Misc	606	419	54	8	1	1	157	1246
	NewPrairiePress	2	4	1	1	0	0	1	9
	Reference	1399	3754	2223	442	79	15	1861	9773
	ResearchConsultation	8	18	21	5	2	1	25	80
	Reserves	36	69	14	1	0	0	35	155
	Technical	413	518	122	27	6	0	232	1318
	Unknown	20	19	3	0	0	0	21	63
	TOTAL	3355	5431	2532	493	89	17	2690	14607

Table 5: READ Scale Ratings vs. Question Type by Count

READ Scale Ratings vs. Question Type: Row-wise (Question Type) Proportions of Transcripts									
		READ							
		1	2	3	4	5	6	Unknown	TOTAL
Question Type	Building	0.53	0.26	0.02	0.00	0.00	0.00	0.19	1
	Circulation	0.32	0.43	0.09	0.01	0.00	0.00	0.15	1
	Copyright	0.00	0.57	0.07	0.07	0.00	0.00	0.29	1
	Directional	0.49	0.24	0.03	0.00	0.00	0.00	0.24	1
	KAPI	0.14	0.43	0.29	0.00	0.00	0.00	0.14	1
	KREx	0.25	0.50	0.13	0.00	0.00	0.00	0.13	1
	Misc	0.49	0.34	0.04	0.01	0.00	0.00	0.13	1
	NewPrairiePress	0.22	0.44	0.11	0.11	0.00	0.00	0.11	1
	Reference	0.14	0.38	0.23	0.05	0.01	0.00	0.19	1
	ResearchConsultation	0.10	0.23	0.26	0.06	0.03	0.01	0.31	1
	Reserves	0.23	0.45	0.09	0.01	0.00	0.00	0.23	1
	Technical	0.31	0.39	0.09	0.02	0.00	0.00	0.18	1
	Unknown	0.32	0.30	0.05	0.00	0.00	0.00	0.33	1
	TOTAL	0.2297	0.3718	0.1733	0.0338	0.0061	0.0012	0.1842	1

Table 6: READ Scale Ratings vs. Question Type by Row-wise Proportions

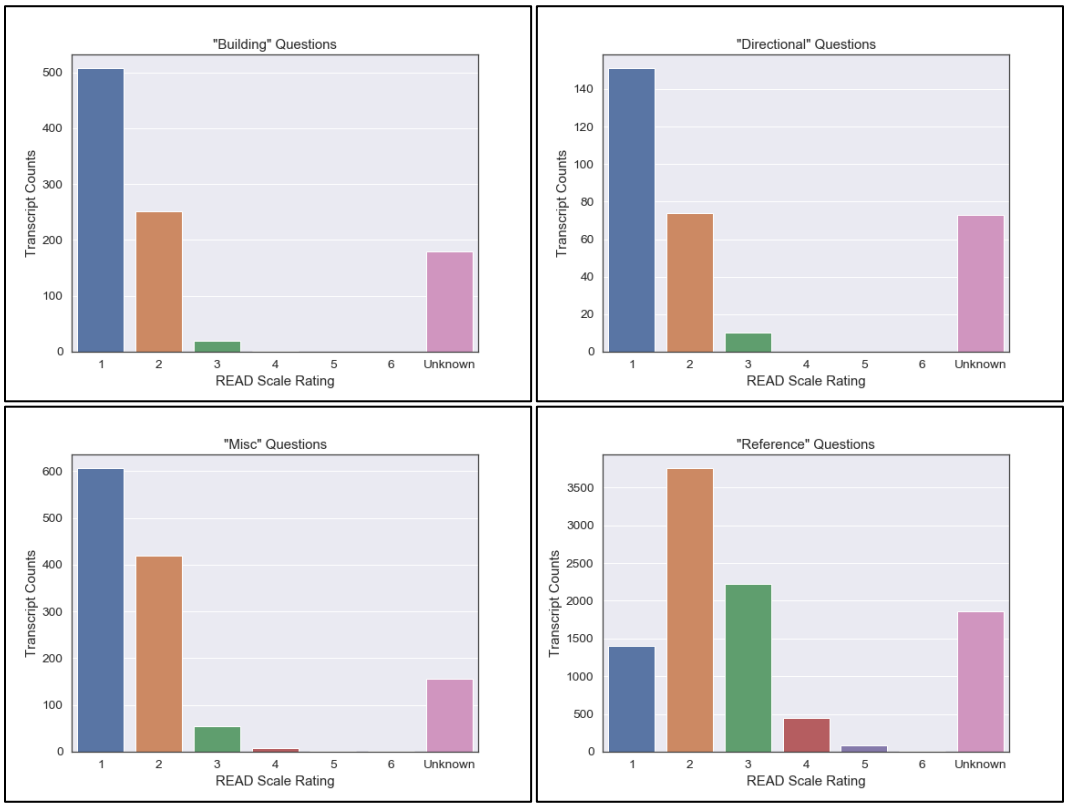

Figure 8: Select READ Scale Rating vs. Question Type Plots

Patron Tokens

The natural language data present in the KSUL dataset's VRS transcripts offer an incredible depth of both challenges and opportunities for the purposes of developing predictive models. After numerous data explorations, two key outcomes emerged.

Even with strict text-processing rules, the underlying data is still likely to contain significant quantities of statistical noise that will erode the quality of any modelling structures that are built upon the data. For example, the distribution of the quantity of natural language data supplied by individual patrons in individual interactions is heavily skewed. The most verbose patron's interaction contains 1061 space-delimited tokens, the least verbose patron's transcript contains a single token, and the average number of tokens is approximately 73 (see Figure 9).

Descriptive Info: Patron Text	
Observations	14604
Average Tokens	72.70
Std.Dev	65.84
Minimum Tokens	1
25%	32
50%	54
75%	91
Maximum Tokens	1061

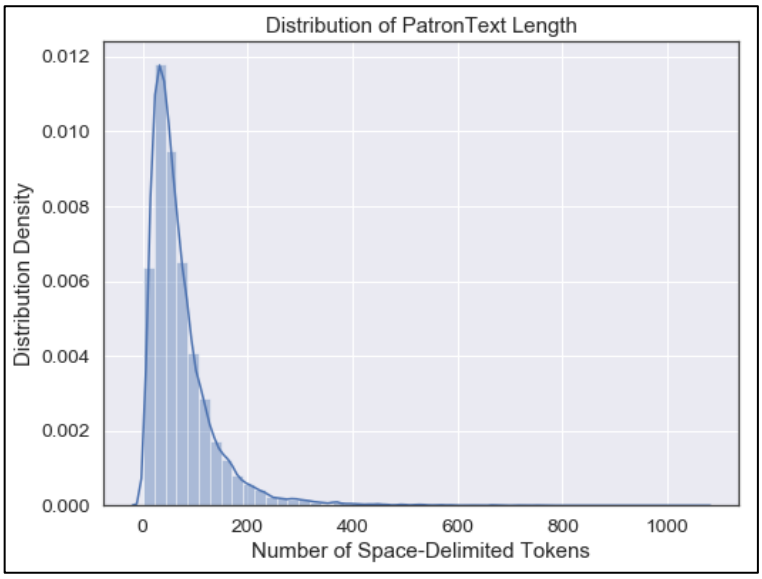

Figure 9: Distribution of Patron Supplied Tokens

For the purposes of predictive modelling in this context, all of the patron-supplied text was truncated to only contain the first ten or twenty tokens (see METHODOLOGY section). Therefore, for the purposes of exploring the data, the first twenty tokens of patron-supplied text were sufficient in most cases for examining the data and looking for patterns. Although truncating the data eliminates noise and detracting data from exceptionally long VRS transcripts, exceptionally short VRS transcripts present a different form of noisy and challenging data. In the two following examples, one patron simply states a call-number (physical location code for library materials) without seeming to engage in any further conversation and the other patron is potentially trolling or pranking the VRS service (see Figure 10). While both examples provide data of marginal value, neither can be unilaterally viewed as useless for predictive modeling since they both represent authentic VRS patron engagement. These and other similar

transcripts were retained in the dataset and might be a source of statistical noise when the data are input into any models.

In addition to a few exceptionally short or uninformative VRS transcripts, the dataset is dominated by tokens that do not necessarily carry much semantic or discriminative value on their own. When VRS transcripts are tokenized and vectorized as a bag-of-words, these issues are illuminated fully. In Table 6, SET#1 shows the 30 most frequently used terms from the first 20 tokens supplied by patrons in their VRS interactions. Most of those 30 terms represent the most commonly used words in the English language and are not by themselves helpful in understanding the conceptual contents of a VRS transcript.

Short Transcript Example #1

['12:22', 'patron', 'P96.742.T89 2010']

['12:22', 'staff', 'Hi!']

['12:22', 'staff', 'Are you wanted to know where this is located?']

Short Transcript Example #2

['6:22', 'patron', 'do you have condoms?']

['6:22', 'staff', "Hi! We sure don't."]

['6:23', 'patron', '']

Figure 10: Parsed VRS Transcript

For the subsequent sets (SET#2, SET#3, SET#4, SET#5), the changes in the size of the token dictionary ('Observation Count') and descriptive statistics are highly sensitive to the rules used to govern which individual words/terms are included or excluded from the data. The 'Selection Parameters' in Table 6 are defined as such:

- Minimum Token Size: the minimum character-length required for any given term to be counted.
- Ngram: Whether to look at terms individually, in pairs, or large combinations for tokenizing.

- Max Doc Frequency: The maximum *percentage* of VRS transcripts in which a token may be present (i.e., document frequency).
- Min Doc Frequency: The minimum *count* of VRS transcripts in which a token may be present.
- Metric: Whether to use raw counts or TF-IDF weightings (Rajaraman & Ullman, 2011).

Strictly looking at the first 30 terms in any given list does not necessarily indicate that high-value words further down in the frequency ranking will not have a strong or positive impact on any part of the modelling process. However, a high preponderance of tokens that do not meaningfully contribute to understanding or representing the data may lead to excessively high-dimensional datasets and models that are overfit to noisy and irrelevant data.

With the application of more discriminatory rulesets, as represented by SET#3 and SET#4, the high-end of the aggregate token lists show that terms like “book” and “article” receive significantly more weight than they do in SET#1 and SET#2. Further, the subtle differences in rank-order between SET#3 and SET#4 suggest that using raw counts and TF-IDF metrics produce slightly different lists of rank-ordered tokens. If any processing rules were to be imposed later on in the modelling process, this finding demonstrates that simply using one metric to justify how the KSUL dataset is truncated may not sufficiently capture the most valuable tokens for modelling purposes.

Importantly, SET#5 demonstrates the problem with attempting to use word combinations, or N-grams, as tokenized data. Although using N-grams of varying size offers the potential to better represent textual content by maintaining syntactic order, using large N-grams like those in SET#5 results in a massive increase in dimensionality and sparsity in the dataset. For SET#5 in particular, the token dictionary is approximately twice the size of the token dictionary of SET#1, yet the total count of the highest rank token in SET#5 is only a small fraction of SET#1’s highest ranked token.

COUNTS AND WEIGHTED FREQUENCIES OF TOKENS ACROSS SETS

	SET#1	SET#2	SET#3	SET#4	SET#5
Selection Parameters					
Minimum Token Size	1	3	3	3	3
Ngram	1-gram	1-gram	1-gram	1-gram	5-gram to 10-gram
Max Doc Frequency	Any	Any	50%	50%	Any
Min Doc Frequency	Any	Any	2	2	2
Metric	Count	Count	Count	TF-IDF	Count
Descriptive Measures					
Observation Count	15347	14898	6208	6208	32167
Mean	19.22	13.64	29.70	0.00	3.13
Std.Dev	249.35	142.64	180.04	0.00	5.92
Minimum	1	1	2	0.000004	2
25%	1	1	2	0.000065	2
50%	1	1	4	0.000115	2
75%	3	3	11	0.000288	3
Maximum	18756	9960	5332	0.042561	302

Top 30 Term/N-Gram	Term	Count	Term	Count	Term	Count	Term	TF-IDF	Term	Count
	i	18756	the	9960	for	5332	you	0.0426	i was wondering if you	302
	the	9960	for	5332	you	5022	for	0.0418	i am trying to find	264
	a	9257	you	5022	and	4410	and	0.0361	i am looking for a	253
	to	8504	and	4410	hello	3838	have	0.0339	hello i am looking for	207
	for	5332	hello	3838	have	3599	hello	0.0337	hi i was wondering if	201
	hi	5075	have	3599	can	2996	can	0.0301	hi i am looking for	194
	you	5022	can	2996	that	2377	book	0.0257	i have a question about	182
	and	4410	that	2377	book	2302	that	0.0256	hello i was wondering if	176
	is	4314	book	2302	find	2223	find	0.0249	was wondering if you could	172
	am	4164	find	2223	library	2179	library	0.0248	hello i am trying to	170
	hello	3838	library	2179	how	2113	how	0.0244	i was wondering if you could	160
	in	3747	how	2113	there	2099	there	0.0237	hi i am trying to	155
	have	3599	there	2099	was	1854	was	0.0221	hello i have a question	149
	of	3576	was	1854	looking	1838	help	0.0212	hi i have a question	141
	it	3313	looking	1838	this	1798	looking	0.0210	i m looking for a	133
	on	3026	this	1798	help	1762	article	0.0202	i m trying to find	132
	can	2996	help	1762	article	1651	thank	0.0200	hi i m trying to	125
	do	2595	article	1651	thank	1516	this	0.0197	hi i m looking for	114
	that	2377	thank	1516	but	1483	need	0.0186	hi i was wondering if you	107
	book	2302	but	1483	need	1480	but	0.0184	i am looking for the	104
	m	2255	need	1480	trying	1394	trying	0.0176	if you could help me	103
	if	2234	trying	1394	about	1326	thanks	0.0167	wondering if you could help	100
	find	2223	about	1326	are	1238	wondering	0.0165	was wondering if you could h elp	94
	library	2179	are	1238	from	1210	about	0.0165	i have a quick question	89
	how	2113	from	1210	wondering	1185	are	0.0158	i am looking for this	88
	there	2099	wondering	1185	thanks	1176	from	0.0154	i am looking for a book	88
	my	2087	thanks	1176	could	1094	could	0.0152	i was wondering if you could help	88
	was	1854	could	1094	with	1074	get	0.0148	wondering if you could help me	88
	looking	1838	with	1074	research	1069	research	0.0146	am looking for a book	88
	an	1825	research	1069	not	1054	question	0.0146	is there a way to	87

Table 7: Counts and Weighted Frequencies of Tokens across Sets

While N-grams may not ultimately be the best choice for the purposes of data processing and representation for modelling, they can be extremely helpful in identifying common linguistic structures, which can then be used to inform the construction of ontologies and fixed representations of the data.

In the KSUL dataset, one easy way to begin looking for patterns in the way library patrons chat in VRS environments is to simply pick keywords and start reading through small windows of adjacent text. In Figure 11, a simple script revealed the contexts in which the keyword “article” appears in patron-

supplied text. By reading through printouts like these, it is relatively easy to get a conceptual sense of the other vocabulary used by VRS patrons when discussing specific keywords and terms.

Figure 11: Sample VRS Transcripts Containing "article"

Going one step further, using the same approach as before to identify high-frequency tokens, it is relatively easy to identify N-grams associated with target keywords. In Table 7, a subset of the patron-supplied text, itself a subset of the whole KSUL dataset, is identified by finding VRS transcripts that contain a specific keyword or string. Then, the texts are tokenized into N-grams ranging from 3 to 10 combinations of space-delimited tokens, counted, and ranked according to total frequency across all relevant documents.

Using this approach reveals that there are hundreds of observations of patrons using very consistent language to describe their needs in a VRS environment. With respect to the keyword "article", many common utterances appear:

- "for an article"
- "find an article"
- "a journal article"

These common phrases all connect to a common theme: The patron is looking for an article to fulfill an information need. For the keyword “floor”, patrons use terms like “3rd floor” and “third floor”; terminology which is conceptually identical but computationally distinct (see Table 7).

Selection Parameters								
VRS Keyword	**"article"**		**"floor"**			**"book"**		
Minimum Token Size	1		3			3		
Ngram	3-gram to 10-gram		3-gram to 10-gram			3-gram to 10-gram		
Max Doc Frequency	Any		Any			Any		
Min Doc Frequency	2		Any			2		
Metric	Count		Count			Count		
Descriptive Measures								
Observation Count	9013		2547			11629		
Mean	4.08		3.13			4.02		
Std.Dev	8.20		4.46			8.58		
Minimum	2		2			2		
25%	2		2			2		
50%	2		2			3		
75%	3		3			253		
Maximum	226		93			5332		
	N-Gram	**Count**	**N-Gram**	**Count**		**N-Gram**	**Count**	
Top 30 N-Gram Phrases	am trying to	226	the third floor	93		am looking for	253	
	am looking for	217	the 3rd floor	79		looking for a	240	
	i am trying	177	on the third	67		for a book	227	
	i am trying to	175	on the third floor	65		i am looking	224	
	i am looking	175	the 4th floor	55		i am looking for	208	
	trying to find	172	on the 3rd	55		was wondering if	195	
	i am looking for	163	on the 3rd floor	54		i was wondering	181	
	for an article	128	there is a	39		looking for a book	171	
	you help me	108	on the 4th	39		i was wondering if	155	
	find an article	104	on the 4th floor	39		i have a	121	
	am trying to find	101	the first floor	36		am looking for a	121	
	i need to	97	send someone up	32		i looking for	119	
	looking for an	96	on third floor	30		wondering if you	117	
	help me find	95	the fourth floor	30		i am looking for a	103	
	i trying to	91	please send someone	28		am trying to	103	
	looking for an article	90	someone up to	26		was wondering if you	100	
	a journal article	87	am on the	26		do you have	100	
	can you help	86	i am on	25		a book i	97	
	an article that	83	please send someone up	24		find a book	96	
	i am trying to find	82	i am on the	24		a book that	95	
	to find a	80	send someone up to	23		to check out	93	
	i was wondering	79	on the fourth	21		i want to	92	
	to find an	78	on the fourth floor	21		if you have	90	
	i have a	77	on the first	21		am looking for a book	88	
	was wondering if	76	on the first floor	21		in the library	86	
	trying to access	74	to third floor	18		out a book	84	
	to find an article	69	the second floor	18		i am trying to	82	
	an article from	67	on 3rd floor	17		i am trying	82	
	have a question	64	floor of hale	17		i need to	80	
	i looking for	64	of the library	16		to find a	79	

Table 8: N-Gram Counts for Transcripts with Target Keyword

For most individual keywords and potential tokens, the exploratory process above should be relatively intuitive to anybody, regardless of their domain knowledge in libraries. However, librarians with more domain knowledge can also use these approaches to identify and collate other terms and tokens that may be present in the dataset. For example, when looking at the entire set of patron-supplied text data, URLs and call numbers (location codes for physical library materials) stand out as examples of text-strings that are highly varied but may in aggregate represent common concepts.

For instance, many patrons may inquire about how to find a book and share the specific call number they found in the library's online catalog (see Table 8). As individual identifiers, these call numbers are probably not meaningful for the purposes of describing a VRS transcript, but, in aggregate, they all represent a situation where a patron may already know about the existence of a specific item and may need directional assistance in finding the item. Conceptually, this is different from a patron who has not already identified information sources and needs help searching for books in the first place.

With respect to the URLs that patrons copy and paste into the VRS chat window, the same line of logic may apply (see Table 8). Depending on how the raw texts are tokenized, URLs may be represented differently. If the tokenization step separates strings using whitespaces, then URLs would be kept entirely intact. If the tokenization step separates strings using whitespaces and non-alphanumeric characters (the default in Scikit-Learn), then URLs will be split into large lists of tokens. In the latter case, it is conceivable that URLs with common roots and directory structures will contribute to the modelling process in a positive way. However, it is just as likely that meaningful URL tokens will be excluded from any transformed dataset if the tokens are not prevalent enough. In which case, it may be critical for a modelling process to include procedures for identifying the presence of URLs and manually inserting an engineered feature to represent that data. A similar process may also be necessary for other common texts such as DOI numbers.

URLs	Counts		Call Numbers	Counts
chat.downloads.s3-us-west-1.amazonaws.	154		HM22.F8	4
chat.downloads.s3.amazonaws.com/	107		LD2668 .D5	4
searchit.lib.ksu.edu/primo_library/lib	62		ML3531 .G46	3
search.proquest.com.er.lib.k-state.edu	45		LD2668 .R4	3
www.lib.k-state.edu/	36		LB885.H626	2
guides.lib.k-state.edu/	33		GV1796.H8	2
www.sciencedirect.com/science/article/	31		LD2668.D5	2
catalog.lib.ksu.edu/vwebv/	24		BR517.H37	2
ksu-primo.hosted.exlibrisgroup.com:170	24		QL673 .H25	2
k-state-primo.hosted.exlibrisgroup.com	23		HF5439.C6	2
www.ncbi.nlm.nih.gov/pubmed/	17		E184.P7	2
books.google.com/	14		F319.O7	2
apps.lib.k-state.edu/databases/categor	14		QL463 .I65	2
apps.lib.k-state.edu/databases/	11		PS3554.E4425	2
login.er.lib.k-state.edu/	11		QL458 .I575	2
www.sciencedirect.com.er.lib.k-state.e	11		PS217.T7	2
www.google.com/	11		HD9397.U54	2
searchit.lib.ksu.edu/	10		HN90.S6	2
catalog2.lib.ksu.edu/vwebv/	10		LB2341 .K63	2
onlinelibrary.wiley.com.er.lib.k-state	9		T55 .A5	2

Table 9: High Frequency URLs and Call Numbers, Top 20

Using these processes of inspecting keywords, terms, phrases, and N-grams, demonstrates the variety of ways that patrons' linguistic patterns and preferred vocabulary can be identified within the KSUL dataset. Although the underlying data can likely never be represented in a purely clean or tidy way, there are sizeable blocks of the data that lend themselves well to value-added feature engineering and data transformations. In particular, the identified processes and patterns can serve as the foundation for constructing a simple ontological model which should, hypothetically, add value to the final representation of the data and to performance of machine learning models.

All of the processes and scripting used to investigate and explore the KSUL dataset are documented in APPENDIX II – HIGH LEVEL DATA EXPLORATION.

METHODOLOGY

In order to effectively test and evaluate whether the inclusion of ontology classes and domain knowledge into machine learning processes is valuable, an experimental design was constructed to systematically test many different modelling processes. Beginning with a corpus of strings supplied by the VRS patrons (*S*), the raw data was processed through a series of modelling steps that altered and transformed the data, produced unsupervised learning models, trained a neural network binary classifier, and generated model performance metrics (see Figure 12). This approach allowed for the direct comparison of every individual modelling process and analysis of the impact on model performance associated with different modelling parameters.

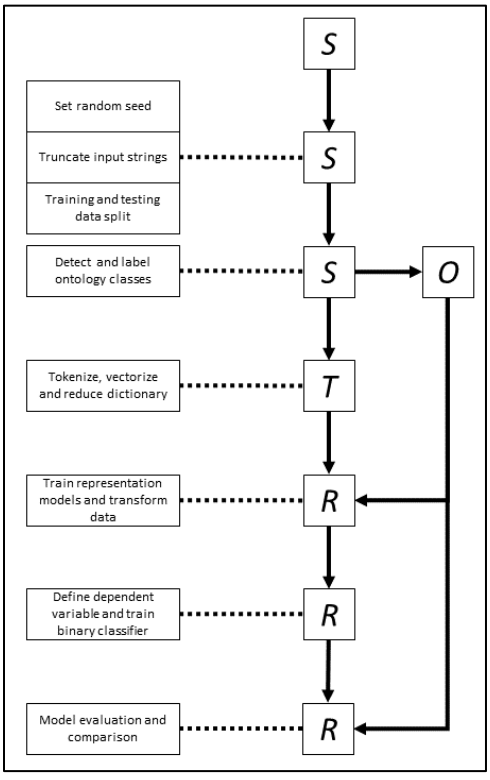

Figure 12: Overall Modelling Process

SET RANDOM SEED

For each model tested, several of the algorithms employed rely on some form of random initialization. The models using LDA representations rely on randomly assigned topic-document and word-topic distributions. The neural-net binary classifier and the models which used Doc2Vec for data representation require randomly initiated weight matrices. In order to control these random elements, a random seed value was fixed at the beginning of each modelling process. Furthermore, to evaluate whether model performance was a consequence of converging to sub-optimal local minima, all modelling processes were run using twenty different random seeds.

- *Options – 0, 1 … 19.*

TRUNCATE INPUT STRINGS

Since the primary motivation behind these modelling processes is to develop predictive models, the patron-supplied strings were truncated. When looking at the space-delimited number of tokens in all patron-supplied texts, the 25th-percentile of patron-supplied texts had 32 tokens. Two arbitrary cutoffs were then set at 10 tokens and 20 tokens in length. This reflects the reasoning that a useful predictive model needs to be able to "see" the first handful of tokens of a patron's inquiry in order to make a prediction or classification. However, the selection of cutoff points of 10 and 20 is arbitrary and other heuristic cutoffs may be more or less appropriate in different circumstances.

- *Option 1 – Limit patron-supplied strings (S_{text}) to first 10 space-delimited tokens.*
- *Option 2 – Limit S_{text} to first 20 space-delimited tokens.*

TRAINING AND TESTING DATA SPLIT

The truncated strings from the KSUL dataset were then split into *training* and *testing* subsets. Since the KSUL dataset, and all service operations in general, exist in a time-dependent public services context, the dataset was sorted chronologically. The 2,000 most recent samples were held aside for testing and external validation, and all of the approximately 15,000 preceding samples were used for defining dictionaries, model training, and internal validation.

DETECT AND LABEL ONTOLOGY CLASSES

The primary goal of building an ontology for this dataset is to provide a robust, structured way to map predictable statements provided by patrons to known sets of questions, behaviors, and needs. The hypothesis is that an ontology, if structured and implemented in a focused way, can enhance the predictive power of a model in a statistically significant way and provide library managers with a mechanism for ensuring that specialized domain knowledge is reflected within the model in a meaningful way.

The instructions laid out in the guide authored by Noy and McGuinness (2001) served as the primary tool for developing an ontology for KSUL's VRS services and dataset. Although research such as that performed by Noy and McGuinness (2001); Ibrahim and Ahmad (2010); Kozareva (2014); and Tanev (2014) demonstrates a variety of ways in which sprawling and intricate ontologies can be developed, the predictive modelling objectives pertaining to the KSUL dataset do not warrant the development of a complex ontology.

Instead, the ontology developed for the KSUL dataset is intentionally kept relatively shallow and simple. This ensures that the ontology remains accessible to external readers and that it can be extended by other librarians in the future.

Once the ontology was finalized, patron-supplied strings (S_{train} , S_{test}) were scanned using regular-expression (REGEX) scripts to detect and label manifestations of ontology class representations. This produced matrices for both the training and testing data (O_{train} , O_{test}) which one-hot encoded the absence or presence of every ontology super-class and core-class for every sample in the dataset. There were three options available for using the ontology data within the modelling process. These options represent the extent to which the detection of ontology sub-classes in the patron-supplied strings is incorporated into model training and prediction.

- *Option 1 – No ontology data is incorporated into modelling.*
- *Option 2 – Ontology data pertaining only to core-classes is incorporated into modelling.*
- *Option 3 – Ontology data pertaining to both core-classes and super-classes is incorporated into modelling.*

Define Ontology Structure

In the context of producing predictive models for the KSUL dataset, developing an ontology's core structure is relatively simple. The ontological super-classes at the top of the hierarchy represent the target dependent variable used for the final stage of the predictive model: READ Scale ratings. Alternative modelling processes could explore building ontologies tailored to different dependent variables (e.g., sentiment), but that is beyond the scope of this project. Then, logically, the ontological sub-classes at the bottom of the hierarchy consist of the raw natural language strings supplied by VRS patrons. Core-classes bridge the gap between the super-classes and sub-classes in the ontology's hierarchy (see Figure 13).

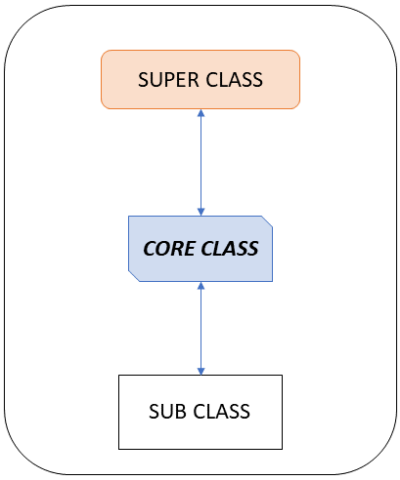

Figure 13: Simple Ontology Hierarchy

Using Noy and McGuiness' (2001) "combination approach" of building ontologies from the "middle" outwards, a few core-classes need to be identified. As a starting point, common and predictable sets of questions and inquiries that librarians engage with are an intuitive place to begin building an ontology. Experienced librarians may be able to develop an ontology based on a general awareness of the common questions presented by library patrons.

On the extreme end of the spectrum, staff at library service desks are frequently asked simple questions like, "The stapler is broken, do you have another?" or "Where is the printer?" The former exemplifies a category of questions that are so ubiquitous across institutions that that type of question has achieved a certain degree of ironic fame within academic libraries (Vance, 2013). Questions like these form an easy starting point for defining core-classes that represent, conceptually, common groupings of questions that tend to rely on predictable terminology.

Using "Where is the printer?" as an example, a core-class may be labelled as "Printing". Then, all sub-classes, representing raw text strings present in the patron-supplied messages, may be defined as the text strings and REGEX patterns that conceptually reflect the idea of a question relating to library printers.

The core-class may then be explicitly related to an intermediary or super-class. Candidate super-classes for a core-class like “Printing” could be informed by the “Question Type” metadata labels already extant in the dataset. These classes, in turn, could be mapped to a final super-class representing the READ Scale rating most commonly associated with a particular “Question Type” (see Figure 14).

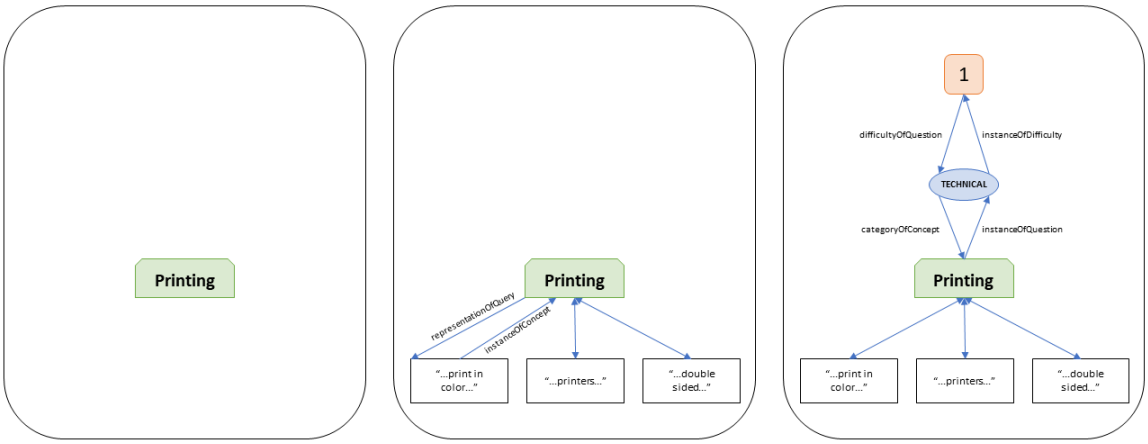

Figure 14: Ontology Development Process for "Printing" Core Class

While this approach to developing an ontology was initially intuitive, problems emerged and required a reworking of how the ontology hierarchy was defined. The first major problem was that although the “Question Type” labels were easily identified in the dataset and intuitively understood, each of the types did not connect cleanly with individual READ Scale ratings. Second, while “Printing” may have been an intuitive concept to categorize within an ontology containing “Question Type” classes, other concepts that may have manifested from the raw string data in the sub-classes did not intrinsically map directly to any one particular “Question Type”. For instance, a VRS patron who states “…trouble finding an article…” may be having technical difficulties with library databases (“Technical” question type) or not have the information literacy skills necessary to search library databases in general (“Reference” question type). Consequently, an ontology using READ Scale ratings and “Question Types” as class definitions is

prone to result in convoluted structures that are sub-optimal for both computer modelling and human attempts to extend or interpret the ontology (see Figure 15).

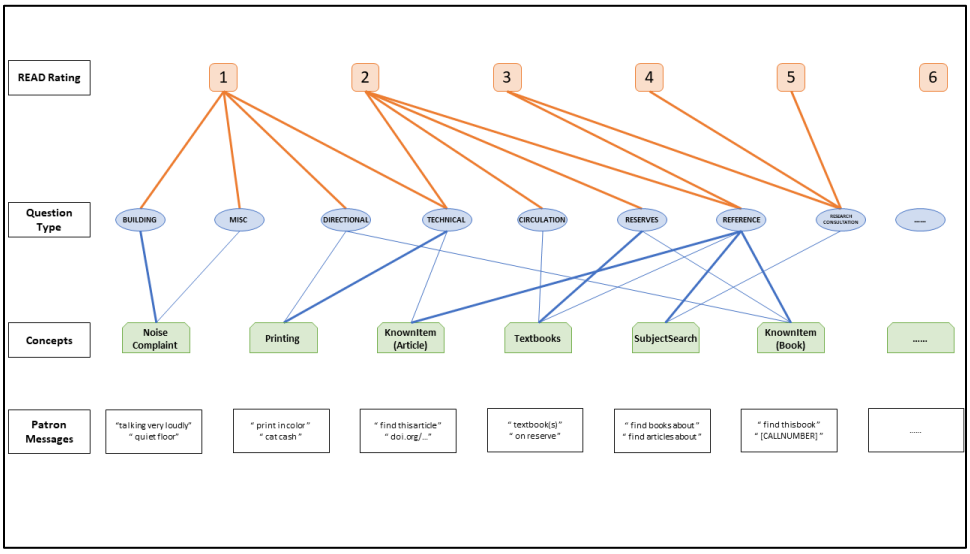

Figure 15: Example of Fully Connected Ontology Using READ Scale Ratings and Question Type Classes

A simple approach to remedying these problems was to simply replace the many READ Scale ratings and "Question Type" classes with a unified set of binary super-classes: "EASIER" and "HARDER". As core-classes and sub-classes were explored, refined, and defined, core-classes with an average READ Scale rating of less than 2.0 were mapped to "EASIER" and classes with average ratings greater than or equal to 2.0 were mapped to "HARDER" (see Figure 16).

This approach did not guarantee that individual samples were exclusively tagged with "EASIER" or "HARDER" super-classes and some samples were tagged with both depending on the strings present in the raw data, how those strings were defined in sub-classes, and how those classes were mapped to core-classes and super-classes. Although this approach did not completely eliminate the ambiguity associated with the first ontology draft (Figure 15), the simplified structure more readily enabled the definition of sub-classes that could be mapped meaningfully to higher level classes in the ontology.

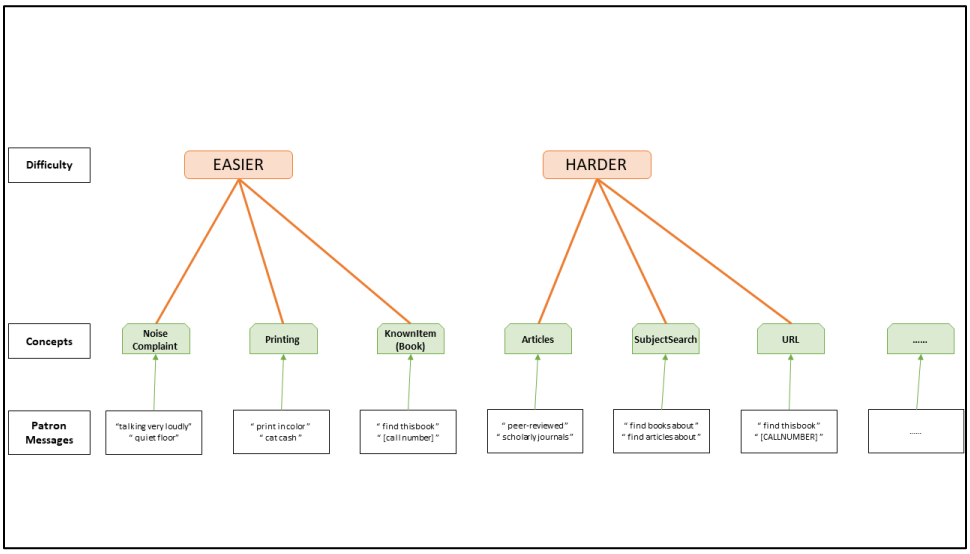

Figure 16: Example of Simplified Ontology Structure

Explore and Define Ontology Classes

Once the overall ontology structure was in place, appropriate core-classes needed to be defined by iteratively exploring common strings and patterns present in the raw data, defining sub-classes in terms of substrings or REGEX patterns, and evaluating whether these patterns reliably detect concepts related to the concepts represented by the core-classes. As already mentioned, librarian expertise and domain knowledge can be used to define many core-classes. For both external observers and library specialists, inspiration for appropriate core-classes and sub-class definitions can also come from individual term and n-gram frequency counts. The metadata and form-field options available to VRS operators can provide clues about the types of predictable, and therefore ontologically definable, patron-interactions that may manifest in the dataset (see Figure 17 and Figure 18). Although this metadata was not ultimately useful for the purposes of developing an ontology structure or modelling in general, it provides suggestions for what types of patterns to look for in the raw text data.

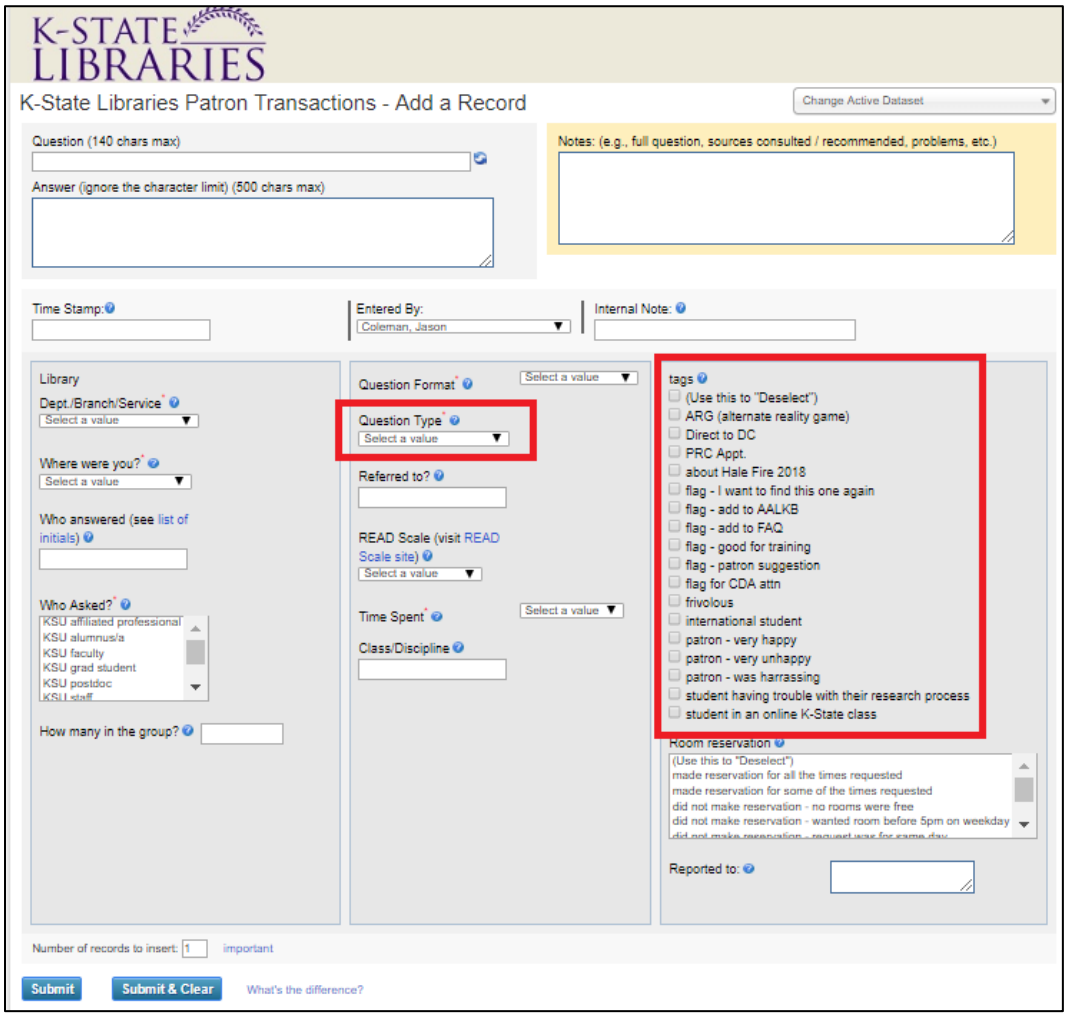

Figure 17: KSUL LibAnalytics Submission Form, Emphasis: Question Type and Tags.

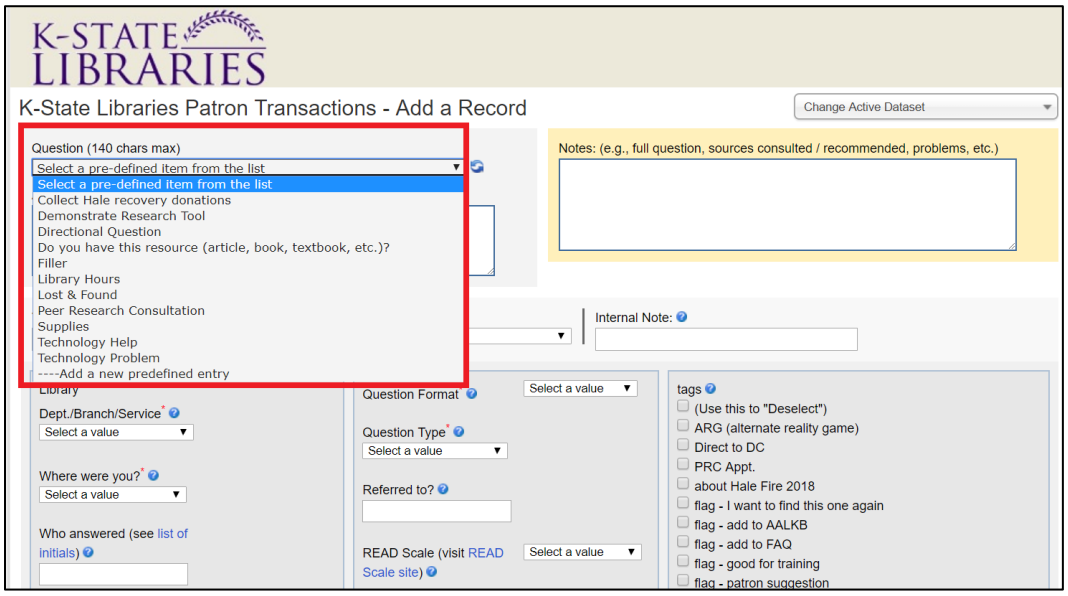

Figure 18: KSUL LibAnalytics Submission Form, Emphasis: Predefined entries.

As candidate core-classes and sub-classes were identified, testing to ensure that sub-class definitions consistently mapped to core-class concepts was a critical challenge. Although it may be common to use stemmed version of words in some text modelling processes, the nature of the data in the KSUL dataset did not lend itself to this approach. For example, although the core-class for "Printing" VRS interactions may intuitively depend on the word "print", that particular term may identify false-positives if defined as a sub-class of "Printing". The same processes used during the data exploration phase can be readily employed to investigate the utility and potential pitfalls of using any particular definition of a sub-class (see Figure 19).

looking for the **print** copy of Marvelous Worlds

am able to utilize the **print**ing services at

is there a color **print**er in the library

Do we have a **print** copy of it in

Figure 19: Sample VRS Transcripts Containing "print"

In order to avoid conflicting and confounding sub-class definitions, core-classes were instead mapped to more robust sets of sub-class definitions that made use of REGEX patterns, unstemmed terminology, and unambiguous phrases and n-grams. For this project, a limited number of core-classes were identified along with robust sub-class definitions for the purposes of creating and incorporating an ontology into the modelling processes. However, in practice, the ontology could be readily extended to include more core-classes and different definitions for sub-classes as appropriate for different institutions. Table 9 provides a high-level overview of all super-classes and core-classes, and limited examples of sub-class definitions.

For full ontology structure and conceptual definition of core-classes, see APPENDIX III – ONTOLOGY STRUCTURE.

super-classes	core-classes	sub-classes
tags / labels attributed to individual samples	*tags / labels attributed to individual samples*	*substring or REGEX patterns to be detected in raw user-supplied texts.*
tagEASIER	tagPRINTING	print in color
		\Win color\W
	tagSCANNER	scanner
		\Wscan\W
	tagOPEN	opens{0,1}\W
		will be open
	tagTEXTBOOKS	on reserve
		course reserve
	tagQUIET	really loud
		stop talking
	tagLIBLOCATION	first floor
		1st floor
	tagLIBMATHPHYS	math/physics library
		math and physics library
	tagLIBWEIGEL	weigel
		wiegel
	tagLIBHALE	hale library
		(?<!help\s)hale
	tagLIBSTACKS	library stacks
		the stacks
	tagKNOWNITEMBOOK	this book
		[a-z]{1,2}\d{2,4}\s{0,1}\.[a-z]\d{1,}
tagHARDER	tagARTICLES	peer.{1}review
		journal article
	tagEVIDENCEBASED	evidence.based
		kinesiology
	tagJUVENILE	juvenile literature
		re.escape(juv. lit)
	tagCURRICULUM	curriculum materials
		curriculum books
	tagKNOWNITEMARTICLE	doi\.org\S+
		this article
	tagREFERENCE	articles{0,1}\sabout
		books{0,1}\sabout
	tagURL	http\S+
		www\S+

Table 10: Final Ontology Outline with Limited Sub-Classes Defined

Detect and Tag Ontology Classes

After developing and fully defining the ontology, all patron-supplied strings (S_{train} , S_{test}) were programmatically scanned using REGEX pattern searching to detect every individual sub-class. If an individual sample contained a string matching a sub-class definition, then that sample was tagged with both the sub-class' parent core-class and super-class. Subsequently, two matrices were generated (O_{train} , O_{test}) representing one-hot encoded representations for all samples. Although it is possible that any given individual sample may have contained multiple instances of sub-classes that mapped to the same core-class, or multiple core-classes that mapped to the same super-class, all duplications were ignored and all core-classes and super-classes were represented with binary 0/1 values.

TOKENIZE, VECTORIZE, AND REDUCE DICTIONARY

After patron-supplied strings were truncated and then processed for ontology-class detection, the strings were tokenized and vectorized, and represented in matrix form. Then, depending on which modelling option was being tested, either all alphanumeric tokens were retained in the vectorized data or rules were applied to remove some alphanumeric tokens from the matrix. Rules included removing tokens which did not meet a minimum character length, meet a minimum document frequency, or were not ranked highly enough according to counting and weighted tf-idf metrics. The primary objective of applying these rules, rules which largely follow common text processing techniques, was to radically reduce the dimensionality and sparsity of data (Pedregosa et al., 2011; Weiss, Indurkhya, & Zhang, 2015).

Following tokenization, all samples (S_{train} , S_{test}) were transformed into respective matrices (T_{train} , T_{test}) in which each sample was represented by a dictionary-length vector of term counts. The dimensionality of these matrices was *samples x dictionary-length*. The size and contents of the dictionary

used to vectorize and count term frequencies for individual samples was dictated by the following modelling choices.

- *Option 1 – No truncation, unfiltered*
    - Strings tokenized by all non-alphanumeric characters.
    - All vocabulary retained in dictionary.
- *Option 2 – Rule-based truncation and filtering of text tokens*
    - Strings tokenized by all non-alphanumeric characters.
    - Only alphanumeric tokens with a minimum length of 3 characters retained.
    - Only tokens with a document-frequency greater than 2 retained.
    - Only the top 3000 tokens from either of the following lists retained:
        - Top 3000 tokens by total count frequency across all strings.
        - Top 3000 tokens by mean TF-IDF weighting across all strings.

The following exemplifies how different processing techniques affected the tokenization and eventual vectorization of patron-supplied strings:

- *Raw string*

  “hi where is P105 .V913 stack? which floor?”
- *Option 1 – Example*:

  ['hi', 'where', 'is', 'p105', 'v913', 'stack', 'which', 'floor']
- *Option 2 – Example*

  ['where', 'stack', 'which', 'floor']

TRAIN REPRESENTATION MODELS AND TRANSFORM DATA

Following the detection of ontology classes in the raw strings (S_{train} , S_{test}), the creation of ontology matrices (O_{train} , O_{test}), and the tokenization and matrix representation of raw strings (T_{train} , T_{test}), machine learning algorithms were used to 'learn' new representations of the data with reduced dimensionality. In both cases, these algorithms transformed the bag-of-words matrices (T_{train} , T_{test}) into their final matrix representations (R_{train} , R_{test}). Additionally, if prior steps in the modelling process called for the inclusion of ontology classes, the data contained in the ontology matrices (O_{train} , O_{test}) were incorporated into the training of the machine learning algorithms and final representation of the data.

- *Options 1 – LDA Representation*
- *Option 2 – Doc2Vec Representation*

Although both options are unsupervised learning methods for representing data, LDA and Doc2Vec require different approaches for incorporating fixed domain knowledge in the form of ontology classes into the data modelling and representation process. Also, the LDA and Doc2Vec algorithms as implemented in a variety of programming packages can be tuned with a wide variety of hyper-parameters such as batch training sizes, learning rates, decay, sampling methods, and many more. Since the purpose of this research was not to conduct an exhaustive grid-search to fine tune the parameters of individual models, most hyper-parameter settings were set to the default configuration in the Gensim implementation of both LDA and Doc2Vec unless otherwise noted.

LDA Representation

The LDA algorithm, as introduced by Blei et al. (2003), requires the definition of three core hyper-parameters: $\alpha, \beta, \text{and } k$. The α and β parameters reflect the dirichlet distribution assumptions pertaining to the proportion of latent topics per document and the proportion of words per topic, respectively, and

k represents the imposed or assumed number of latent topics. Iterative testing of LDA topic-models using the T_{train} revealed perplexity metrics indicating that k = 75 was a reasonable choice for the number of latent topics. LDA topic-models trained on T_{train} exhibited a sudden decrease in perplexity indicating that at values above k = 75, the LDA model may have been overfitting the data (see Figure 20).

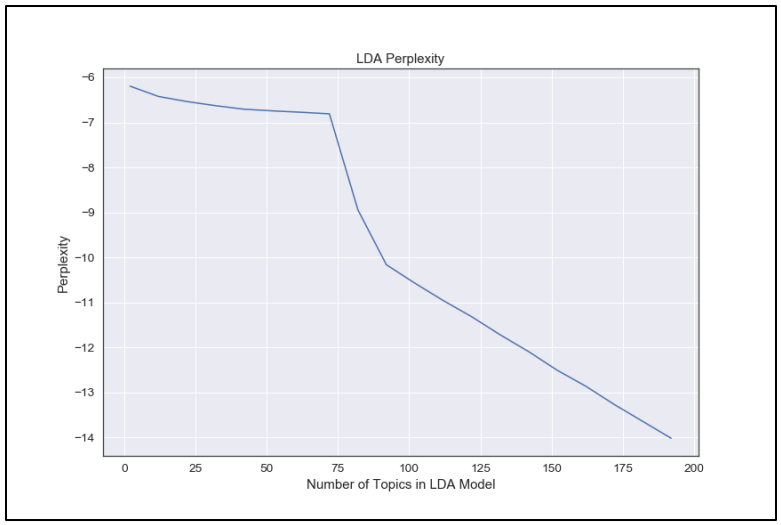

Figure 20: Evaluation of LDA Models' Perplexity

Following the selection of a value of k=75, the values for α and β were both then defined as equaling $\frac{1}{k}$ or approximately 0.013. These parameters were selected to ensure that any given sample would be associated with a relatively small number of latent topic distributions (θ) and that each topic would consist of a relatively small proportion of words (w) (see Figure 21).

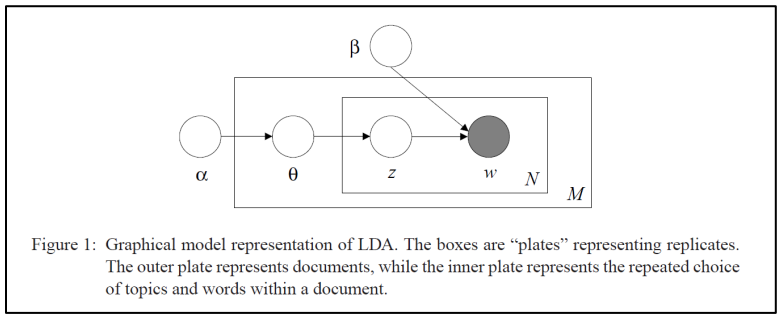

Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

Figure 21: Blei et al., 2003

Once the final LDA topic-model was trained, the tokenized datasets (T_{train} , T_{test}) were transformed according to the topic-word distribution vectors, resulting in two latent topic matrices (L_{train} , L_{test}) in which every sample was represented by a fixed-length vector of size *k.*

Finally, when early modelling steps indicated that ontology classes need to be incorporated into modelling, then L_{train} and O_{train}, and L_{test} and O_{test} were concatenated into final representation matrices (R_{train} , R_{test}). Other research and development into LDA models includes methods for incorporating tags and domain knowledge directly into the machine learning processes (Ramage et al., 2009; Zhu et al., 2006). However, these approaches were deemed unwieldy in comparison to simply concatenating the data (see Figure 22).

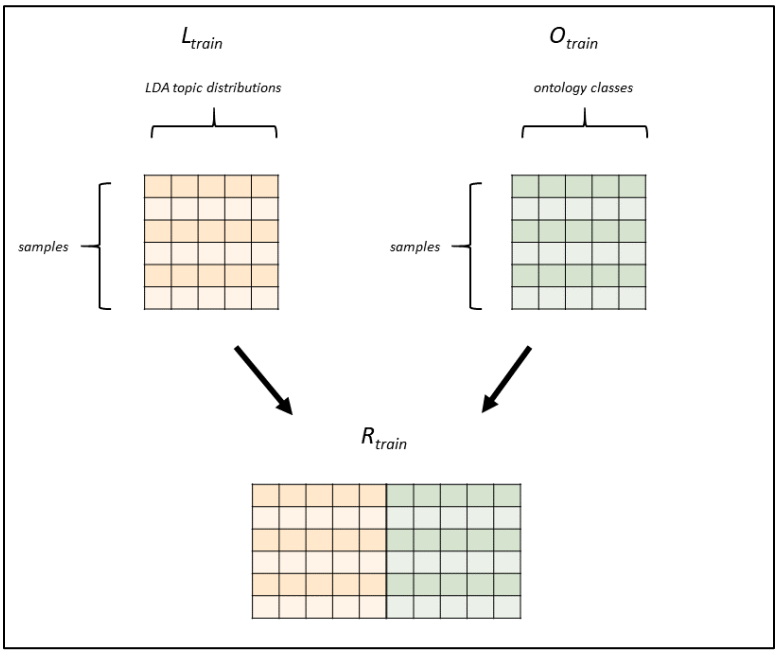

Figure 22: Concatenation of LDA and Ontology Data into Final Representations

Doc2Vec Representation

The Doc2Vec algorithm is a method for representing entire documents as fixed-length vectors. Every document, or sample in this case, is given a unique identifier and a random vector representation. Although randomly initiated, these vectors represent the weight matrices in a shallow neural network model. In the original paper, the Google researchers present two different model formulations for inferring document vector representations: PV-DM ('distributed memory') and PV-DBOW ('distributed bag of words') (Le & Mikolov, 2014). These document modelling processes are analogous to the CBOW and Skip-Gram model architectures for the Word2Vec algorithm (Mikolov et al., 2013).

Extensively testing the difference in these modelling architectures was beyond the scope of this project. However, since the PV-DBOW model explicitly ignores word order, it is more directly comparable to the alternative LDA model. Additionally, researchers have suggested that the PV-DBOW model (see Figure 23) outperforms the PV-DM model in a variety of tests (Lau & Baldwin, 2016).

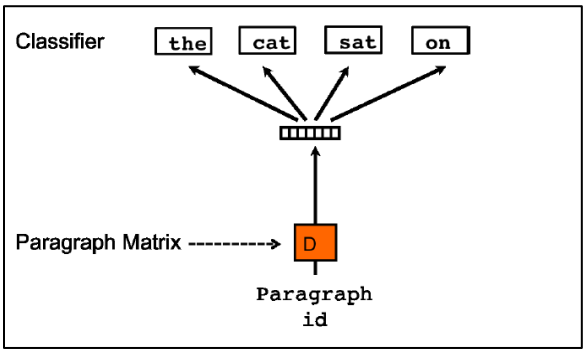

Figure 23: Distributed Bag of Words version of Paragraph Vector (PV-DBOW)

(Le & Mikolov, 2014)

The Gensim implementation of Doc2Vec also enables the training of what are described as "tags" in the Python package (Řehůřek & Sojka, 2010). Therefore, in addition to training the neural network model to find vector representations of individual documents, the same process can be applied to aggregated labels. For the purposes of this project, that means that data encoded in the O_{train} could be integrated directly into the Doc2Vec algorithm's learning process. By default, the training stage of the Doc2Vec model learns vectors of weights, representing individual samples, and weights connecting the hidden layer of the neural network with the word-tokens in the output layer. These vectors are learned in concert throughout training. For this project, ontology class labels could also be incorporated into the training stage (see Figure 24). Consequently, during training, the model was expected to produce a neural network model in which the weight-matrix connecting the hidden layer and the output layer adequately captured the relationships between samples with similar word distributions and samples with shared ontology labels.

Following training of the Doc2Vec model, the output weights were frozen. At this stage, all of the training and testing samples (T_{train} , T_{test}) were fed into the Doc2Vec model as unseen samples. Vectors for each sample were randomly initialized and then inferred and updated over the course of 100 training iterations. During this process, the only aspect of the Doc2Vec model that updated the individual samples'

vector representation. This process yielded final representation matrices for both training and testing samples (R_{train} and R_{test}). Each individual sample was represented by a fixed-length vector of 75 units. This vector representation also reflected the size of the hidden layer in the Doc2Vec model.

The selection of 75-unit vectors was arbitrary and intended to make the representations generated by the Doc2Vec model comparable to the topic-model representations generated by the LDA model. However, whereas the LDA model explicitly encoded ontology class labels by concatenating the LDA (L_{train}, L_{test}) and ontology (O_{train}, O_{test}) representations, the Doc2Vec models' final vector representations of individual samples did not explicitly include ontology labels. Instead, these labels were implicitly encoded in the vector representations as a consequence of how the Doc2Vec model was trained (see EXPERIMENTAL RESULTS section).

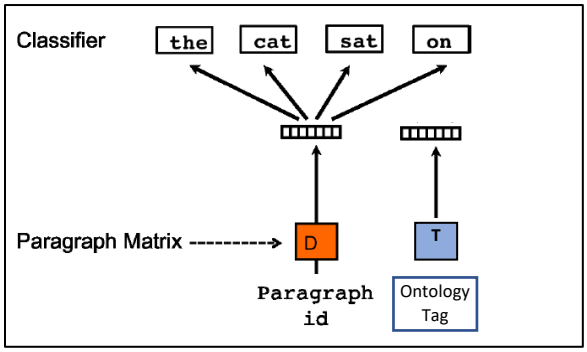

Figure 24: PV-DBOW including training of ontology tag vectors

DEFINE DEPENDENT VARIABLE AND TRAIN AND EVAULATE BINARY CLASSIFIER

After all of the raw text data were processed, tokenized, tagged, and transformed using LDA/Doc2Vec, the final vector representations were used to train and evaluate a binary classifier. For the binary dependent variable, ordinal READ Scale ratings were split and converted into a binary representation. Since earlier data exploration did not indicate an obvious breakpoint in the distribution

of READ Scale ratings, two different options were used to test different modelling processes. Any samples in both the training and testing sets for which READ Scale ratings were unavailable were ignored.

- *Option 1 – READ Scale rating 1 labelled as "easy" (0) and READ Scale ratings 2, 3, 4, 5, & 6 labelled as "hard" (1)*
- *Option 2 – READ Scale rating 1 & 2 labelled as "easy" (0) and READ Scale ratings 3, 4, 5, & 6 labelled as "hard" (1)*

A shallow neural network classifier was then trained using the R_{train} data and associated easy/hard labels. The model architecture consisted of an input layer, a single hidden layer with ten nodes, and a single-node sigmoid activated output layer. This neural network model was then used to generate probability predictions for each training and testing sample's *R* vector represenation. The neural network classifier was used due to its relative computational efficiency and ability to capture non-linear relationships manifest in the data. Alternative modelling choices could have made use of support vector machines or non-linear regression models, but that was unnecessary in the context of building models for the KSUL dataset.

MODEL EVALUATION & COMPARISON

Following the modelling parameters outlined in the preceding sections, 960 unique modelling processes were evaluated (see Table 10). Final predictions for both R_{train} and R_{test} were evaluated by finding the receiver operating characteristic (ROC) and associated area-under-the-curve (AUC). Again, samples without valid READ Scale ratings were ignored. The AUC metric, which evaluates the discriminative power of a classification model, was used as the primary performance metric for both training and testing datasets. Further, the training and testing datasets were sub-divided into two subsets;

one representing samples that were tagged as containing at least one instance of a sub-class from the ontology, and one representing samples that did not contain any instances of ontology sub-classes.

Parameter Name	Label	Modelling Options																			
Random Seed	RAND	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
String Truncation	TRUNC	First10										First20									
Ontology - Core Classes	O-Core	FALSE										TRUE									
Ontology - Core + Super Classes	O-Core+Super	FALSE										TRUE (only if O-Core == True)									
Dictionary Truncation	DICT	ALL										LIMITED									
Data Representation	REPRESENT	LDA										Doc2Vec									
Dependent Variable	READ	READ1vs2										READ2vs3									

Table 11: Modelling Parameters

After running all models and collecting performance metrics, all models were paired and filtered into pairs of models representing 'neighbors.' Neighboring models represented modelling processes in which only one modelling parameter was different (see Table 11). Only single-parameter neighboring models were identified; all other pairs were removed from the resulting comparisons table (see Table 12). For each pair of models, the difference (Δ) between the AUC performance metrics was calculated and recorded (see EXPERIMENTAL RESULTS section).

Model ID	TRUNC	O-Core	O-Core+Super	DICT	REPRESENT	READ	AUC (training)	AUC (test)
8	First10	TRUE	FALSE	ALL	LDA	READ_1_vs_2	0.7043	0.6581
16	First10	FALSE	FALSE	ALL	LDA	READ_1_vs_2	0.6917	0.6527
9	First10	TRUE	FALSE	LIMITED	LDA	READ_1_vs_2	0.7011	0.6506
17	First10	FALSE	FALSE	LIMITED	LDA	READ_1_vs_2	0.6856	0.6601
10	First10	TRUE	FALSE	ALL	D2V	READ_1_vs_2	0.7257	0.6737
18	First10	FALSE	FALSE	ALL	D2V	READ_1_vs_2	0.7224	0.6846
11	First10	TRUE	FALSE	LIMITED	D2V	READ_1_vs_2	0.7053	0.6807
19	First10	FALSE	FALSE	LIMITED	D2V	READ_1_vs_2	0.6959	0.6835

Table 12: Example of Pairs of Neighboring Models

Combination ID	Model ID #1	Model ID #2	Common Parameter	Delta AUC (training)	Delta AUC (test)	Δ AUC (training) *(tagged with classes)*	Δ AUC (test) *(tagged with classes)*	Δ AUC (training) *(not tagged)*	Δ AUC (test) *(not tagged)*
7651	8	16	O-Core	0.0126	0.0054	0.1213	0.0635	-0.0216	-0.0130
8602	9	17	O-Core	0.0155	-0.0095	0.1283	0.0621	-0.0194	-0.0314
9552	10	18	O-Core	0.0032	-0.0109	0.0753	0.0212	-0.0182	-0.0238
10501	11	19	O-Core	0.0094	-0.0028	0.0906	0.0385	-0.0158	-0.0143

Table 13: Example of Comparison Table

All of the processes and scripting used to implement experimental design and store results in flat CSV files are documented in APPENDIX IV – EXPERIMENTAL DESIGN, RESULTS, ANALYSES*.*

EXPERIMENTAL RESULTS

Examination of the pair-wise comparisons of all neighboring models produced valuable statistical inferences for each of the core modelling parameters. The results indicated that incorporating domain knowledge in the form of ontology classes and labels into the modelling process can, in general, have a positive and statistically significant impact on predictive modelling. This was a rejection of the original null hypothesis outlined at the beginning of the research project. Further evidence that the impact of ontology labels was not spurious was demonstrated in the analysis of representation vectors (R_{train} , R_{test}) generated from modelling processes using the Doc2Vec algorithm.

MODEL AND PARAMETER PERFORMANCE

Ontology Parameters – Core Classes

The results related to models incorporating ontological labels showed an overall positive and significant impact on predictive modelling performance. However, upon close inspection of the results, this positive impact came with a few extremely important caveats (see Table 13). With respect to the testing subset, the O-Core modelling parameter, representing the incorporation of core-class labels into the modelling and representation of the data, had a small but significant impact on predictive performance (Average Δ AUC = +0.0029, p-value < 0.001). The practical significance for O-Core's impact on the AUC performance metric was relatively small compared to the parameters READ and TRUNC. Furthermore, when viewing t-scores as indicators of the magnitude of a parameter's significance, the O-Core parameter was still relatively small compared to others.

However, when segmenting the testing data into two subsets representing samples that contain instances of the ontology's sub-classes and samples that do not ("Tagged" and "Untagged" respectively in Table 13), more practically significant results were seen. For tagged samples, there was an average

increase of +0.0628 on modelling performance for models in which the O-Core parameter was enabled. Additionally, the O-Core parameter's t-score was dramatically increased (t-score = 41.7515). This was in contrast to the reduced practical and statistical significance of all other modelling parameters.

For untagged samples, the results were also noteworthy. When the O-Core parameter was enabled, the results showed that there was a practically and statistically significant negative impact on modelling performance for untagged samples (Average Δ AUC = -0.0235, p-value < 0.001). When considered with the prior observation, this indicates that the incorporation of the ontology's core-classes into the modelling process dramatically improved the performance of predictive models for tagged documents while decreasing predictive performance for untagged documents. One possible explanation for this is that during training, the neural network classifier was limited to a small, fixed number of training iterations. As such, the predictive model may capture the relationships between samples tagged with ontology classes more quickly than it captures the relationships between untagged samples. Owing to the imposed limitations on training iterations, this could have resulted in the model overfitting its predictions for tagged samples and under-fitting its predictions for untagged samples.

			Δ AUC TEST DATA					Δ AUC TEST DATA - TAGGED WITH ONTOLOGY CLASSES					Δ AUC TEST DATA - UNTAGGED WITH ONTOLOGY CLASSES				
Parameter	*Parameter Setting*	*Models Compared*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*
DICT	Default: ALL Alternative: LIMITED	480	-0.0029	0.0136	4.6681	0.0000	***	0.0024	0.0160	3.3062	0.0010	**	-0.0054	0.0170	6.9891	0.0000	***
O-Core	Default: FALSE Alternative: TRUE	320	0.0029	0.0065	8.0513	0.0000	***	0.0628	0.0269	41.7515	0.0000	***	-0.0235	0.0111	37.9836	0.0000	***
O-Core+Super	Default: FALSE Alternative: TRUE	320	0.0001	0.0055	0.3110	0.7560		0.0008	0.0095	1.5240	0.1285		-0.0002	0.0068	0.4581	0.6472	
RAND	[0 .. 19]	9120	0.0003	0.0120	2.0867	0.0369	*	-0.0004	0.0141	2.6184	0.0088	**	0.0005	0.0148	3.0413	0.0024	**
READ	Default: READ 1vs2 Alternative: READ 2vs3	480	-0.0234	0.0123	41.5811	0.0000	***	-0.0043	0.0209	4.5538	0.0000	***	-0.0332	0.0162	44.8339	0.0000	***
REPRESENT	Default: LDA Alternative: D2V	480	0.0108	0.0138	17.0472	0.0000	***	-0.0001	0.0179	0.0828	0.9341		0.0158	0.0171	20.3240	0.0000	***
TRUNC	Default: First10 Alternative: First20	480	0.0355	0.0145	53.6624	0.0000	***	0.0183	0.0205	19.6203	0.0000	***	0.0332	0.0174	41.7511	0.0000	***

** P-value < 0.05*
*** P-value < 0.01*
**** P-value < 0.001*
DF = (Models Compared) - (Parameters) -1

Counts of Samples Labelled with READ Ratings		
Training Data	10162	
Testing Data	1753	
	First10	**First20**
Training - Tagged	1832	3033
Training - Untagged	8330	7129
Testing - Tagged	340	549
Testing - Untagged	1413	1204

Table 14: Model Test Performance, Emphasis Parameter T-Scores

Ontology Parameters – Super Classes

In contrast to the notable impact that the O-Core parameter had on model performance, the results associated with the O-Core+Super parameter indicate that super-classes, as defined in the ontology structure, did not have a meaningful impact on the modelling process and performance metrics (see Table 14). The logic of the experimental design was such that O-Core+Super could only be incorporated into modelling processes for which O-Core was also enabled. Thus, the results associated with O-Core+Super strictly reflect the value of the ontology super-classes "tagEASIER" and "tagHARDER".

The original reasoning for including these super-classes into the structure of the ontology and modelling processes was based on the hypothesis that using high-level binary labels would ultimately help the binary-classification model produce more accurate predictions. However, when looking at the testing subsets, this does not appear to have been the case. For the whole testing subset, including the subsets with tagged and untagged samples, the incorporation of super-classes into the modelling process had a negligible and statistically insignificant impact on performance metrics. Furthermore, on average, the incorporation of super-class labels had less of an impact on modelling performance than the modelling processes' random initialization (RAND).

			Δ AUC TEST DATA					Δ AUC TEST DATA - TAGGED WITH ONTOLOGY CLASSES					Δ AUC TEST DATA - UNTAGGED WITH ONTOLOGY CLASSES				
Parameter	*Parameter Setting*	*Models Compared*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*
DICT	Default: ALL Alternative: LIMITED	480	-0.0029	0.0136	4.6681	0.0000	***	0.0024	0.0160	3.3062	0.0010	**	-0.0054	0.0170	6.9891	0.0000	***
O-Core	Default: FALSE Alternative: TRUE	320	0.0029	0.0065	8.0513	0.0000	***	0.0628	0.0269	41.7515	0.0000	***	-0.0235	0.0111	37.9836	0.0000	***
O-Core+Super	Default: FALSE Alternative: TRUE	320	0.0001	0.0055	0.3110	0.7560		0.0008	0.0095	1.5240	0.1285		-0.0002	0.0068	0.4581	0.6472	
RAND	[0 .. 19]	9120	0.0003	0.0120	2.0867	0.0369	*	-0.0004	0.0141	2.6184	0.0088	**	0.0005	0.0148	3.0413	0.0024	**
READ	Default: READ 1vs2 Alternative: READ 2vs3	480	-0.0234	0.0123	41.5811	0.0000	***	-0.0043	0.0209	4.5538	0.0000	***	-0.0332	0.0162	44.8339	0.0000	***
REPRESENT	Default: LOA Alternative: D2V	480	0.0108	0.0138	17.0472	0.0000	***	-0.0001	0.0179	0.0828	0.9341		0.0158	0.0171	20.3240	0.0000	***
TRUNC	Default: First10 Alternative: First20	480	0.0355	0.0145	53.6624	0.0000	***	0.0183	0.0205	19.6203	0.0000	***	0.0332	0.0174	41.7511	0.0000	***

** P-value < 0.05*
*** P-value < 0.01*
**** P-value < 0.001*
DF = (Models Compared) - (Parameters) -1

Counts of Samples Labelled with READ Ratings		
Training Data	10162	
Testing Data	1753	
	First10	**First20**
Training - Tagged	1832	3033
Training - Untagged	8330	7129
Testing - Tagged	340	549
Testing - Untagged	1413	1204

Table 15: Model Test Performance, Emphasis O-Core+Super and RAND Parameters

Controlling Parameters

The modelling options available for some of the tested parameters resulted in performance metrics that perfectly matched expectations. For example, the selection of the TRUNC parameter option, representing whether a model is limited to the first 10 or the first 20 word-tokens, had a dramatic impact on modelling performance. This parameter had an average positive impact of +0.0358 on modelling performance for the training subset and +0.0037 on the test subset (see Table 15). This is perfectly logical since, at its core, this modelling parameter represents how much data was being allowed into the model: more data translated to better predictions. The same logic explains the negative impact of the DICT modelling parameter. The results for the DICT parameter showed an average negative impact of -0.0079 for the training subset and -0.0017 for the testing subset. Although the practical impact was relatively small, the results were statistically significant. Again, these results are intuitive and logical since the DICT parameter, when enabled, removed low-frequency terms and reduced the amount of data available throughout the rest of the modelling process.

Random Seeds

The results from the RAND modelling parameter, simply representing the randomization seed for the machine learning processes (LDA, Doc2Vec, and neural net classifier), suggest that further tuning of model hyper-parameters is necessary before any modelling processes are implemented in practice. In the results for the training and testing subsets, the RAND parameter had a very small but statistically significant impact on modelling performance. This suggests that the random initialization of different modelling processes may have resulted in the LDA, Doc2Vec, and neural net classifier models converging to different local minima with respect to data representation and predictive modelling. With specific respect to the neural network classifier, the Python script used explicitly stated that the model had not yet reached convergence (see Figure 25).

```
C:\Users\jw\Anaconda3\lib\site-packages\sklearn\neural_network\multilayer_perceptron.py:562: Converg
enceWarning: Stochastic Optimizer: Maximum iterations (100) reached and the optimization hasn't conv
erged yet.
  % self.max_iter, ConvergenceWarning)
```

Figure 25: Scikit-Learn MLP Classifier Convergence Warning

However, this does not suggest that the individual models were incomparable. Because the pair-wise comparison of each modelling process only included comparisons of models with identical RAND parameter settings, any challenges associated with random initializations and model convergence were equally applicable to any two compared models. Therefore, the results and inferences gleaned from other modelling parameters remain valid.

Other Modelling Parameters

The results related to the remaining modelling parameters, READ and REPRESENT, also demonstrate both a practical and statistically significant impact on model performance. However, similar

to the results related to the O-Core parameter, the true impact of the respective options for these modelling parameters was only evident when investigating the tagged and untagged subsets.

For instance, with respect to samples containing instances of ontology sub-classes, and therefore potentially having been tagged with core-class and super-class labels, there was no meaningful difference between using LDA or Doc2Vec to generate sample vector representations (Average Δ AUC = -0.0001, p-value = 0.9341). Conversely, with respect to untagged test samples, there was a clear, practical, and statistically positive impact on modelling performance associated with using Doc2Vec over LDA (Average Δ AUC = 0.0158, p-value < 0.001). These results reflect average model performance across all pairs of modelling processes, including both models that did incorporate ontology tags into data representations and those that did not. Consequently, this suggests that for tagged samples, the raw data were so consistent and predictable that there was little reason to favor LDA over Doc2Vec for predictive modelling purposes. On the other hand, for untagged samples, the superior performance of modelling processes using Doc2Vec indicates that the Doc2Vec algorithm was better suited than LDA at learning representations and relationships between words and samples.

Similarly, for the READ parameter, the difference between the parameter's options had a greater impact on modelling performance for untagged samples compared to tag samples. Although the average Δ AUC for both tagged and untagged subsets was statistically significant, the magnitude of the impact was an order of magnitude stronger for the untagged samples compared to tagged samples (-0.0332 vs -0.0043).

			Δ AUC TRAINING DATA				Δ AUC TRAINING DATA - TAGGED WITH ONTOLOGY CLASSES				Δ AUC TRAINING DATA - UNTAGGED WITH ONTOLOGY CLASSES			
Parameter	*Parameter Setting*	*Models Compared*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*
DICT	Default: ALL	480	-0.0078	0.0085	20.0827	0.0000	***	-0.0079	0.0084	20.4315	0.0000	***	-0.0078	0.0095	18.1106	0.0000	***
	Alternative: LIMITED																
O-Core	Default: FALSE	320	0.0085	0.0085	17.7549	0.0000	***	0.0750	0.0325	41.2383	0.0000	***	-0.0164	0.0085	34.7242	0.0000	***
	Alternative: TRUE																
O-Core+Super	Default: FALSE	320	0.0006	0.0056	2.0367	0.0425	*	0.0013	0.0072	3.1294	0.0019	**	0.0003	0.0062	0.9810	0.3273	
	Alternative: TRUE																
RAND	[0…19]	9120	-0.0003	0.0073	4.5349	0.0000	***	-0.0004	0.0083	4.0732	0.0000	***	-0.0004	0.0081	4.5160	0.0000	***
READ	Default: READ 1x2	480	0.0046	0.0083	12.1179	0.0000	***	-0.0045	0.0107	9.1940	0.0000	***	0.0110	0.0117	20.6069	0.0000	***
	Alternative: READ 2x3																
REPRESENT	Default: LDA	480	0.0234	0.0170	30.1665	0.0000	***	0.0001	0.0282	0.0809	0.9356		0.0333	0.0186	39.0938	0.0000	***
	Alternative: D2V																
TRUNC	Default: First10	480	0.0357	0.0147	53.2181	0.0000	***	0.0092	0.0243	8.2517	0.0000	***	0.0359	0.0179	44.0345	0.0000	***
	Alternative: First20																

			Δ AUC TEST DATA					Δ AUC TEST DATA - TAGGED WITH ONTOLOGY CLASSES					Δ AUC TEST DATA - UNTAGGED WITH ONTOLOGY CLASSES				
Parameter	*Parameter Setting*	*Models Compared*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*	*Average Δ AUC*	*Std Dev.*	*T-Score*	*P-Value*	*Significance*
DICT	Default: ALL	480	-0.0029	0.0136	4.6681	0.0000	***	0.0024	0.0160	3.3062	0.0010	**	-0.0054	0.0170	6.9891	0.0000	***
	Alternative: LIMITED																
O-Core	Default: FALSE	320	0.0029	0.0065	8.0513	0.0000	***	0.0628	0.0269	41.7515	0.0000	***	-0.0235	0.0111	37.9836	0.0000	***
	Alternative: TRUE																
O-Core+Super	Default: FALSE	320	0.0001	0.0055	0.3110	0.7560		0.0008	0.0095	1.5240	0.1285		-0.0002	0.0068	0.4581	0.6472	
	Alternative: TRUE																
RAND	[0…19]	9120	0.0003	0.0120	2.0867	0.0369	*	-0.0004	0.0141	2.6184	0.0088	**	0.0005	0.0148	3.0413	0.0024	**
READ	Default: READ 1x2	480	-0.0234	0.0123	41.5811	0.0000	***	-0.0043	0.0209	4.5538	0.0000	***	-0.0332	0.0162	44.8339	0.0000	***
	Alternative: READ 2x3																
REPRESENT	Default: LDA	480	0.0108	0.0138	17.0472	0.0000	***	-0.0001	0.0179	0.0828	0.9341		0.0158	0.0171	20.3240	0.0000	***
	Alternative: D2V																
TRUNC	Default: First10	480	0.0355	0.0145	53.6624	0.0000	***	0.0183	0.0205	19.6203	0.0000	***	0.0332	0.0174	41.7511	0.0000	***
	Alternative: First20																

** P-value < 0.05*
*** P-value < 0.01*
**** P-value < 0.001*
DF = (Models Compared) - (Parameters) -1

Counts of Samples Labelled with READ Ratings			
Training Data	10162		
Testing Data	1753		
		First10	**First20**
Training - Tagged		1852	3033
Training - Untagged		8330	7129
Testing - Tagged		340	549
Testing - Untagged		1413	1204

Table 16: Full Comparison of Model Performance with all Subsets

ONTOLOGY ASSESSMENT

In addition to evaluating the predictive performance metrics of modelling processes with respect to individual modelling parameters and decisions, assessment of the ontology structure revealed that, for the most part, the implementation of domain knowledge was sound and meaningfully integrated into the modelling process.

Similar to the analytic approach described by Alshargi et al. (2018), the ontology structure was evaluated using the ontology tag vectors and individual sample vectors that were produced when using Doc2Vec as part of the modelling process. The following parameter options reflect the specific modelling process used to produce the Doc2Vec model and vectors for both tag labels and all individual samples:

- RAND = 0
- TRUNC = First20
- O-Core = True
- O-Core+Super = True
- DICT = ALL
- REPRESENT = Doc2Vec
- READ = READ1vs2

Once all tag and sample vectors were generated from the Doc2Vec model, each of the tag vectors representing the core-class and super-class labels were evaluated using the cosine-similarity metric (see Table 16). For example, the average cosine-similarity between the vector representing “tagPRINTING” and the vector representations of all samples containing an instance of the sub-classes associated with “tagPRINTING” was compared against the average cosine-similarity between “tagPRINTING” and all other

samples. Cosine-metrics closer to 1.0 represented perfect alignment and -1.0 represented perfectly opposed alignment. Following earlier processes, testing and training subsets were evaluated separately.

When inspecting the cosine-similarities between the vector representations of each ontology class against appropriately tagged samples and then against all other samples, the results overwhelmingly showed that the tag vectors for every ontology class were more closely aligned with relevant tagged samples than with other samples. Furthermore, in both the training and testing subsets, the distinction in the cosine-similarity metrics were highly statistically significant for the vast majority of the ontology classes when analyzed using Welch's t-tests (see Table 16). These results, in general, indicated that the ontology structure used in this modelling process was well suited to informing how individual samples were transformed and represented when using the Doc2Vec model.

There were however a few notable exceptions that appeared in the results pertaining to the testing subset. First, Welch's t-tests for the vectors for "tagSCANNER" and "tagTEXTBOOKS" indicated that these vectors were not as strongly aligned with relevant sample vectors as was the case with other core-classes. This may have been part because there were relatively few samples in the testing data to evaluate or perhaps because the ontology class was not appropriately represented by its sub-classes. Additionally, no comparative metrics could be calculated for the vectors representing "tagLIBMATHPHYS", "tagLIBSTACKS", "tagEVIDENCEBASED", and "tagCURRICULUM". For these four classes, there were not enough samples in the testing data to allow for evaluation or comparison.

When comparing the individual vectors for the core-classes directly against the vectors for super-classes, again using cosine-similarity, all but one of the core-class vectors was more closely aligned with its appropriate super-class than not (see Table 17). In the one contrary example, "tagCURRICULUM", it is possible that the assumption that the sub-classes associated with "tagCURRICULUM" should have mapped to the super-class "tagHARDER" was erroneous. However, this particular core-class did not have a robust

set of samples to investigate and may simply have represented a core-class that needed to be removed from the ontology or aggregated with another core-class.

		TRAINING SUBSET						
Tag ID	Ontology Class	vs Tagged Samples	vs All Other Samples	Abs Difference	T-Test	P-Value	Tagged Count	Other Count
0	tagPRINTING	0.5390	0.0494	0.4896	47.1236	0.0000	76	10086
1	tagSCANNER	0.3979	0.0402	0.3577	12.5687	0.0000	24	10138
2	tagHOURS	0.5889	0.0423	0.5466	52.4889	0.0000	184	9978
3	tagTEXTBOOKS	0.3778	0.0445	0.3333	16.7260	0.0000	44	10118
4	tagQUIET	0.6611	0.0426	0.6185	33.9434	0.0000	44	10118
5	tagLIBLOCATION	0.5980	0.0233	0.5747	92.5098	0.0000	177	9985
6	tagLIBMATHPHYS	0.6207	0.1126	0.5081	9.7396	0.0002	6	10156
7	tagLIBWEIGEL	0.6090	0.0961	0.5129	22.1053	0.0000	9	10153
8	tagLIBHALE	0.4884	0.0368	0.4517	83.6027	0.0000	278	9884
9	tagLIBSTACKS	0.5727	0.0337	0.5390	24.0301	0.0000	18	10144
10	tagKNOWNITEMBOOK	0.2949	0.0269	0.2680	32.7301	0.0000	204	9958
11	tagARTICLES	0.4778	0.0223	0.4555	80.2006	0.0000	493	9669
12	tagEVIDENCEBASED	0.6679	0.0784	0.5895	23.2308	0.0000	11	10151
13	tagJUVENILE	0.4584	0.0463	0.4121	14.7950	0.0000	31	10131
14	tagCURRICULUM	0.7394	0.1073	0.6321	13.1597	0.0000	6	10156
15	tagKNOWNITEMARTICLE	0.3087	0.0182	0.2905	25.2322	0.0000	174	9988
16	tagREFERENCE	0.3411	0.0350	0.3061	29.2761	0.0000	168	9994
17	tagURL	0.5861	0.0376	0.5485	54.1812	0.0000	70	10092
18	tagEASIER	0.3182	0.0234	0.2948	66.9482	0.0000	967	9195
19	tagHARDER	0.2695	0.0116	0.2579	60.2071	0.0000	874	9288
		TESTING SUBSET						
Tag ID	Ontology Class	vs Tagged Samples	vs All Other Samples	Abs Difference	T-Test	P-Value	Tagged Count	Other Count
0	tagPRINTING	0.4578	0.0535	0.4044	9.4638	0.0000	8	1745
1	tagSCANNER	0.3965	0.0354	0.3611	3.0498	0.0925	3	1750
2	tagHOURS	0.5242	0.0396	0.4846	17.9573	0.0000	43	1710
3	tagTEXTBOOKS	0.3283	0.0432	0.2851	4.5911	0.0058	6	1747
4	tagQUIET	0.6444	0.0382	0.6062	12.4923	0.0002	5	1748
5	tagLIBLOCATION	0.6021	0.0164	0.5857	34.3638	0.0000	19	1734
6	tagLIBMATHPHYS		0.1161				0	1753
7	tagLIBWEIGEL	0.6765	0.1031	0.5733	16.7105	0.0004	4	1749
8	tagLIBHALE	0.4703	0.0339	0.4365	37.0790	0.0000	62	1691
9	tagLIBSTACKS	0.3901	0.0314	0.3587			1	1752
10	tagKNOWNITEMBOOK	0.2214	0.0261	0.1954	11.4257	0.0000	46	1707
11	tagARTICLES	0.4445	0.0264	0.4181	29.4374	0.0000	86	1667
12	tagEVIDENCEBASED		0.0741				0	1753
13	tagJUVENILE	0.4646	0.0414	0.4231	6.7750	0.0010	6	1747
14	tagCURRICULUM	0.4001	0.1101	0.2900			1	1752
15	tagKNOWNITEMARTICLE	0.3021	0.0206	0.2815	12.4510	0.0000	53	1700
16	tagREFERENCE	0.2801	0.0273	0.2529	9.1854	0.0000	27	1726
17	tagURL	0.5904	0.0431	0.5473	30.3796	0.0000	19	1734
18	tagEASIER	0.2834	0.0218	0.2616	22.1717	0.0000	176	1577
19	tagHARDER	0.2466	0.0169	0.2297	19.9440	0.0000	167	1586

Table 17: Cosine Similarity Comparisons between Tag Vectors and Samples

		SUPER-CLASS			
Tag ID	**Ontology Class**	**vs tagEASIER**	**vs tagHARDER**	**Intended Label**	**Actual Label**
0	tagPRINTING	0.5660	-0.1463	EASY	EASY
1	tagSCANNER	0.6956	-0.1629	EASY	EASY
2	tagHOURS	0.5569	-0.2216	EASY	EASY
3	tagTEXTBOOKS	0.5186	-0.3487	EASY	EASY
4	tagQUIET	0.5359	-0.1135	EASY	EASY
5	tagLIBLOCATION	0.6367	-0.1436	EASY	EASY
6	tagLIBMATHPHYS	0.4581	-0.3888	EASY	EASY
7	tagLIBWEIGEL	0.5451	-0.2620	EASY	EASY
8	tagLIBHALE	0.7822	-0.2093	EASY	EASY
9	tagLIBSTACKS	0.7889	-0.1706	EASY	EASY
10	tagKNOWNITEMBOOK	0.6860	0.0755	EASY	EASY
11	tagARTICLES	-0.1521	0.6006	HARD	HARD
12	tagEVIDENCEBASED	-0.3464	0.2633	HARD	HARD
13	tagJUVENILE	0.0469	0.3815	HARD	HARD
14	tagCURRICULUM	0.0526	-0.0141	HARD	EASY
15	tagKNOWNITEMARTICLE	-0.0352	0.7661	HARD	HARD
16	tagREFERENCE	-0.2234	0.6556	HARD	HARD
17	tagURL	-0.1210	0.8535	HARD	HARD
18	tagEASIER	1.0000	-0.1473		
19	tagHARDER	-0.1473	1.0000		

Table 18: Cosine Similarity Comparisons between Core-Class and Super-Class Vectors

IMPLICATIONS AND CONCLUSIONS

Ontology Structures & Modelling

The key outcome of this research was to show that although automatic natural language processing techniques such as LDA, Doc2Vec, and even more simple measures like TF-IDF metrics are increasingly easy to use and implement, the incorporation of domain knowledge, as represented by a simple ontology structure, can still be a valid and valuable component of predictive modelling processes. In this particular project, the incorporation of a simple ontology structure and class labels into data representations and modelling had a positive, profound, and significant impact on predictive models centered around VRS transcripts in an academic library setting. Further, the simplistic structure of the ontology created for this project also represents a framework from which other librarians may benefit. Specifically, others may be able to extend the scope of the ontology by identifying new core-classes and refining the definitions of sub-classes as appropriate for different VRS datasets at different institutions.

With respect to the ontology structure developed for this project, although the model reflects the nature of VRS interactions at KSUL, the structure is already out of date and may require additional configuration before KSUL service managers can consider implementing the model. Although the pace and frequency with which different academic libraries may need to update ontology structures for the purposes of predictive modelling is likely to be generally slow, KSUL may require additional intervention earlier owing to a significant and disruptive fire that took place in May of 2018 (Hoyt, 2019). As a result of the disruption, it is very likely that the ontology structure developed for this project no longer adequately reflects the full scope of ways in which VRS patrons describe and engage with library services and resources.

Hyper-parameter Tuning

While all of the modelling processes presented in this research ultimately represent predictive models that perform distinctly better than a purely random model, the experimental design of the modelling processes means that no single model can be identified as "the best". By comparing these modelling processes, model parameters can be examined, measured, and evaluated for utility with respect to improving, and only improving, baseline model performance. Before any model is used in a real-life environment, such as a triage-system built into a VRS platform, further extensive model hyper-parameter tuning will be necessary in order to identify optimal modelling processes. This includes more extensive training times for machine learning algorithms, experimenting with various sizes of VRS vector representations, and other hyper-parameters throughout the modelling process.

Operational Considerations

Lastly, a high-level review of modelling performance suggests that libraries with VRS services may be able to benefit greatly from the incorporation of predictive modelling processes into their operations. The models experimented with in this model were built on the assumption that a predictive model designed to automatically route incoming VRS inquiries to different types of operators (see Figure 2) would be valuable and useful to library service managers. This assumption is based on the concerns expressed by researchers who have identified shortcomings in student VRS operators' skills in referring VRS patrons efficiently (Keyes & Dworak, 2017; Lux & Rich, 2016). Furthermore, these models can also be used to inform the development and deployment of automated chatbots and recommender systems that have already been deployed at various institutions (Kane, 2019; University of Oklahoma Libraries, 2019c).

However, before any of the insights or models demonstrated in this paper can be used in a real VRS or predictive-modelling context, service managers must consider strategic operational factors that go beyond the scope of this project. Assuming a model is developed for the purposes of incorporating a predictive model that triages incoming VRS patron inquiries to different levels of VRS operators according

to predicted READ Scale ratings, service managers must decide what the most appropriate dependent variable is in order for any predictive model to adequately align with an institution's strategic and service objectives. What is the appropriate decision-boundary for any given set of dependent variables? Should the model be more prone to false-positives or false-negatives? What is the minimum amount of initial input text that should be solicited from a patron prior to the commencement of a VRS interaction? Prior to actually implementing any predictive models using machine learning, ontologies, or even traditional statistics, questions like these must be addressed by service managers in order to fully leverage these modelling processes.

REFERENCES

Al-Salemi, B., Ab Aziz, Mohd. J., & Noah, S. A. (2015). LDA-AdaBoost.MH: Accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization. *Journal of Information Science*, *41*(1), 27–40. https://doi.org/10.1177/0165551514551496

Alshargi, F., Shekarpour, S., Soru, T., Sheth, A., & Quasthoff, U. (2018). Concept2vec: Metrics for evaluating quality of embeddings for ontological concepts. *ArXiv:1803.04488 [Cs]*. Retrieved from http://arxiv.org/abs/1803.04488

Althoff, T., Clark, K., & Leskovec, J. (2016). Large-scale analysis of counseling conversations: An application of natural language processing to mental health. *Transactions of the Association for Computational Linguistics*, *4*, 463–476. https://doi.org/10.1162/tacl_a_00111

ARL. (2019, January 31). *ARL statistics questionnaire, 2017-2018: Instructions and definitions*. Retrieved from http://www.arlstatistics.org/About/Mailings/stats_2017-18

Baumgart, S., Carrillo, E., & Schmidli, L. (2016). Iterative chat transcript analysis: Making meaning from existing data. *Evidence Based Library and Information Practice*, *11*(2), 39–55. https://doi.org/10.18438/B8X63B

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Boman, C. (2019). An exploration of machine learning in libraries. In J. Griffey (Ed.), *Artificial intelligence and machine learning in libraries* (pp. 21–25). Chicago, IL: ALA TechSource.

Boman, C., Kim, B., Yelton, A., & Griffey, J. (2019). *Artificial intelligence and machine learning in libraries* (J. Griffey, Ed.). Chicago, IL: ALA TechSource.

Bracke, M. S., Brewer, M., Huff-Eibl, R., Lee, D. R., Mitchell, R., & Ray, M. (2007). Finding information in a new landscape: Developing new service and staffing models for mediated information services. *College & Research Libraries*, *68*(3). https://doi.org/10.5860/crl.68.3.248

Bravender, P., Lyon, C., & Molaro, A. (2011). Should chat reference be staffed by librarians? An assessment of chat reference at an academic library using libstats. *Internet Reference Services Quarterly*, *16*(3), 111–127. https://doi.org/10.1080/10875301.2011.595255

Burger, A., Jung-ran Park, & Guisu Li. (2010). Application of reference guidelines for assessing the quality of the internet public library's virtual reference services. *Internet Reference Services Quarterly*, *15*(4), 209–226. https://doi.org/10.1080/10875301.2010.526479

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 22* (pp. 288–296). Retrieved from http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf

Fuller, K., & Dryden, N. H. 2. (2015). Chat reference analysis to determine accuracy and staffing needs at one academic library. *Internet Reference Services Quarterly*, *20*(3/4), 163–181. https://doi.org/10.1080/10875301.2015.1106999

Gerlich, B. K., & Berard, G. L. (2007). Introducing the READ scale: Qualitative statistics for academic reference services. *Georgia Library Quarterly*, *43*(4).

Gerlich, B. K., & Berard, G. L. (2010). Testing the viability of the READ Scale (Reference Effort Assessment Data)©: Qualitative statistics for academic reference services. *College & Research Libraries*, *71*(2), 116–137. https://doi.org/10.5860/0710116

Hoyt, S. M. (2019, May 22). What Hale Library means to K-State students. Retrieved May 29, 2019, from Hale Library Blog website: https://blogs.k-state.edu/hale/2019/05/22/students/

Ibrahim, M., & Ahmad, R. (2010). Class diagram extraction from textual requirements using natural language processing (NLP) techniques. *2010 Second International Conference on Computer Research and Development*, 200–204. https://doi.org/10.1109/ICCRD.2010.71

Iwashita, M., Shimogawa, S., & Nishimatsu, K. (2011). Semantic analysis and classification method for customer enquiries in telecommunication services. *Engineering Applications of Artificial Intelligence*, *24*(8), 1521–1531. https://doi.org/10.1016/j.engappai.2011.02.016

Jacoby, J., Ward, D., Avery, S., & Marcyk, E. (2016). The value of chat reference services: A pilot study. *Portal: Libraries and the Academy*, *16*(1), 109–129. https://doi.org/10.1353/pla.2016.0013

Kane, D. (2019). Analyzing an interactive chatbot and its impact on academic reference services. *ACRL 2019 Proceedings*, 481–492. Cleveland, OH.

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., & Giannopoulou, E. (2007). Ontology visualization methods—a survey. *ACM Computing Surveys*, *39*(4). https://doi.org/10.1145/1287620.1287621

Keyes, K., & Dworak, E. (2017). Staffing chat reference with undergraduate student assistants at an academic library: A standards-based assessment. *The Journal of Academic Librarianship*, *43*(6), 469–478. https://doi.org/10.1016/j.acalib.2017.09.001

King, V., & Christensen-Lee, S. (2014). Full-time reference with part-time librarians. *Reference & User Services Quarterly*, *54*(1), 34–43. https://doi.org/10.5860/rusq.54n1.34

Kohler, E. (2017). What do your library chats say?: How to analyze webchat transcripts for sentiment and topic extraction. *Brick & Click Libraries Conference Proceedings*. Presented at the 17th Annual Brick & Click Libraries Conference, Maryville, Missouri.

Koshik Irene, & Okazawa Hiromi. (2012). A conversation analytic study of actual and potential problems in communication in library chat reference interactions. *Journal of the American Society for Information Science and Technology*, *63*(10), 2006–2019. https://doi.org/10.1002/asi.22677

Kozareva, Z. (2014). Simple, fast and accurate taxonomy learning. In C. Biemann & A. Mehler (Eds.), *Text Mining: From Ontology Learning to Automated Text Processing Applications* (pp. 41–62). https://doi.org/10.1007/978-3-319-12655-5_3

Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *Proceedings of the 1st Workshop on Representation Learning for NLP*, 78–86. https://doi.org/10.18653/v1/W16-1609

Lauren, P., Qu, G., Zhang, F., & Lendasse, A. (2018). Discriminant document embeddings with an extreme learning machine for classifying clinical narratives. *Neurocomputing*, *277*, 129–138. https://doi.org/10.1016/j.neucom.2017.01.117

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International Conference on Machine Learning*, 1188–1196. Retrieved from http://proceedings.mlr.press/v32/le14.html

Liu, Y., Wang, J., & Jiang, Y. (2016). PT-LDA: A latent variable model to predict personality traits of social network users. *Neurocomputing*, *210*, 155–163. https://doi.org/10.1016/j.neucom.2015.10.144

Lux, V. J. 1, & Rich, L. (2016). Can student assistants effectively provide chat reference services? Student transcripts vs. librarian transcripts. *Internet Reference Services Quarterly*, *21*(3/4), 115–139. https://doi.org/10.1080/10875301.2016.1248585

Maloney, K., & Kemp, J. H. (2015). Changes in reference question complexity following the implementation of a proactive chat system: Implications for practice. *College & Research Libraries*, *76*(7), 959–974. https://doi-org.turing.library.northwestern.edu/10.5860/crl.76.7.959

Matteson, M. L., Salamon, J., & Brewster, L. (2011). A systematic review of research on live chat service. *Reference & User Services Quarterly*, *51*(2), 172–189. https://doi.org/10.5860/rusq.51n2.172

Meert, D. L., & Given, L. M. (2009). Measuring quality in chat reference consortia: A comparative analysis of responses to users' queries. *College & Research Libraries*, *70*(1), 71–84. https://doi.org/10.5860/0700071

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv:1301.3781 [Cs]*. Retrieved from http://arxiv.org/abs/1301.3781

Momtazi, S. (2018). Unsupervised latent dirichlet allocation for supervised question classification. *Information Processing & Management*, *54*(3), 380–393. https://doi.org/10.1016/j.ipm.2018.01.001

Morais, Y., & Sampson, S. (2010). A content analysis of chat transcripts in the Georgetown Law Library. *Legal Reference Services Quarterly*, *29*(3), 165–178. https://doi.org/10.1080/02703191003751289

Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology* (Technical Report No. KSL-01-05). Retrieved from Stanford University website:

http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html

Oubounyt, M., Louadi, Z., Tayara, H., & Chong, K. T. (2018). Deep learning models based on distributed feature representations for alternative splicing prediction. *IEEE Access*, *6*, 58826–58834. https://doi.org/10.1109/ACCESS.2018.2874208

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Powers, A. C., Shedd, J., & Hill, C. (2011). The role of virtual reference in library web site design: A qualitative source for usage data. *Journal of Web Librarianship*, *5*(2), 96–113. https://doi.org/10.1080/19322909.2011.573279

Program O. (2019). Program O AI Chatbot - The friendly open source PHP, MySQL, AIML chatbot. Retrieved May 29, 2019, from Program O AI Chatbot website: https://program-o.com

Radford, M. L. (2006). Encountering virtual users: A qualitative investigation of interpersonal communication in chat reference. *Journal of the American Society for Information Science and Technology*, *57*(8), 1046–1059. https://doi.org/10.1002/asi.20374

Rajaraman, A., & Ullman, J. D. (2011). Data mining. In *Mining of Massive Datasets* (pp. 1–17). https://doi.org/10.1017/CBO9781139058452.002

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, *1*, 248–256. Association for Computational Linguistics.

Řehůřek, 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.

RUSA. (2008a). Definitions of reference. Retrieved February 1, 2019, from Reference and User Services Association website: http://www.ala.org/rusa/guidelines/definitionsreference

RUSA. (2008b, September 29). Guidelines for behavioral performance of reference and information service providers. Retrieved May 12, 2019, from Reference and User Services Association website: http://www.ala.org/rusa/resources/guidelines/guidelinesbehavioral

Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., & Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, *163*, 955–971. https://doi.org/10.1016/j.knosys.2018.10.026

Tanev, H. (2014). Learning textologies: Networks of linked word clusters. In C. Biemann & A. Mehler (Eds.), *Text Mining: From Ontology Learning to Automated Text Processing Applications* (pp. 25–40). https://doi.org/10.1007/978-3-319-12655-5_2

University of Oklahoma Libraries. (2019a). Projects in Artificial Intelligence Registry (PAIR): A registry for AI projects in higher ed. Retrieved May 29, 2019, from Projects in Artificial Intelligence Registry (PAIR) website: https://pair.libraries.ou.edu/

University of Oklahoma Libraries. (2019b, January 18). ANTswers: University of California, Irvine Libraries chatbot. Retrieved May 29, 2019, from Projects in Artificial Intelligence Registry (PAIR) website: https://pair.libraries.ou.edu/content/antswers-university-california-irvine-libraries-chatbot

University of Oklahoma Libraries. (2019c, January 18). Library website chatbot. Retrieved May 29, 2019, from Projects in Artificial Intelligence Registry (PAIR) website: https://pair.libraries.ou.edu/content/library-website-chatbot

Vance, J. (2013). Staplercide!: The lives and deaths of academic library staplers. *College & Research Libraries News*, *74*(11). https://doi.org/10.5860/crln.74.11.9041

Wang, X., Gerber, M. S., & Brown, D. E. (2012). Automatic crime prediction using events extracted from twitter posts. In S. J. Yang, A. M. Greenberg, & M. Endsley (Eds.), *Social Computing, Behavioral - Cultural Modeling and Prediction* (pp. 231–238). Springer Berlin Heidelberg.

Ward, D. (2003). Using virtual reference transcripts for staff training. *Reference Services Review*, *31*(1), 46–56. https://doi.org/10.1108/00907320310460915

Waugh, J. (2013). Formality in chat reference: Perceptions of 17- to 25-year-old university students. *Evidence Based Library and Information Practice*, *8*(1), 19–34. Retrieved from https://doaj.org

Weiss, S. M., Indurkhya, N., & Zhang, T. (2015). *Fundamentals of predictive text mining*.

Zhu, X. J., Blei, D., & Lafferty, J. (2006). *TagLDA: Bringing a document structure knowledge into topic models*. University of Wisconsin-Madison Department of Computer Sciences.

APPENDIX I – DATA PREPARATION

https://arch.library.northwestern.edu/concern/parent/xg94hp819/file_sets/xp68kg47r

APPENDIX II – HIGH LEVEL DATA EXPLORATION

https://arch.library.northwestern.edu/concern/parent/xg94hp819/file_sets/n296wz391

APPENDIX III – FULL ONTOLOGY STRUCTURE

Super-Classes	Core-Classes	Sub-Classes	High-Level Definitions
tags / labels attributed to individual samples	*tags / labels attributed to individual samples*	*substring or REGEX patterns to be detected in raw user-supplied texts.*	*Subjective defintion of the broad concepts associated with core-classes and sub-classes in relation to VRS interactions.*
tagEASIER	tagPRINTING	color print	Interactions relating to using library printers and printing services.
		colored print	
		print in color	
		print something in color	
		\Win color\W	
		cat cash	
		printer	
		(?<!3D\s)\bprinting	
		double.{1}sided	
		catcash	
		cat cash	
		add money	
	tagSCANNER	scanner	Interactions related to using library scanners.
		\Wscan\W	
	tagHOURS	open 24/7	Interactions in which users inquire about library building and service hours.
		what time	
		the hours	
		opens{01}\W	
		will be open	
		summer hours	
		library hours	
	tagLIBMATHPHYS	re.escape(Math/Physics Library)	Interactions in which the Math & Physics Library is explicitly identified.
		re.escape(math and physics library)	
		re.escape(Math Physic library)	
		re.escape(math/physics library)	
		re.escape(maths/phys library)	
		re.escape(math & phys library)	
		re.escape(math phys library)	
	tagLIBWEIGEL	weigel	Interactions in which the Weigel Architecture Library is explicitly identified.
		wiegel	
	tagLIBVETMED	vet med	Interactions in which the Vetrinary Medicine Library is explicitly mentioned.
		vetmed	
	tagLIBHALE	Hale Library	Interactions in which Hale Library is explicitly identified.
		(?<!help\s)hale	
	tagLIBSTACKS	Library Stacks	Interactions in which the users explicitly identify the "stacks".
		the stacks	
		in Stacks	
	tagTEXTBOOKS	the reserve	Interactions in which textbooks and course reserve materials and services are mentioned.
		on reserve	
		course reserve	
		reserve textbook	
		have a specific textbook	
		have the textbook	
		have textbook	
		this textbook	
		this text book	
	tagQUIET	quite loud	Interactions in which the user mentions excessive noise or inquires about quiet places in the library.
		super loud	
		really loud	
		very loud	
		stop talking	
		talking on	
		music loud	
		loud	
		talking very	
		talking extremely	
		talking loud	
		quiet floor	
		quiet floor	
		Quiet Zone	
		quiet floors	
		floor to be quiet	
		whisper quietly	
		be quiet	
		floor to be quiet	
	tagLIBLOCATION	first floor	Interactions in which the user mentions specific locations within the library (i.e. Hale Library)
		1st floor	
		second floor	
		2nd floor	
		third floor	
		3rd floor	
		fourth floor	
		4t floor	
		fifth floor	
		5th floor	
		hemisphere room	
		Harry Potter room	
		the hemi	
	tagKNOWNITEMBOOK	a-z{12}\d{24}\s{01}\.a-z\d{1}	Interactions in which a user identifies a specific, individual book.
		this book	

Super-Classes	Core-Classes	Sub-Classes	High-Level Definitions
tags / labels attributed to individual samples	*tags / labels attributed to individual samples*	*substring or REGEX patterns to be detected in raw user-supplied texts.*	*Subjective defintion of the broad concepts associated with core-classes and sub-classes in relation to VRS interactions.*
tagHARDER	tagARTICLES	peer.{1}review	Interactions in which users ask about accessing, finding, or discovering journal articles in general.
		journal article	
		scholarly article	
		scholarly journal	
		scholarly article	
		peer reviewed	
		re.escape(peer-reviewed)	
		peerreviewed	
		scholarly	
		articles	
	tagEVIDENCEBASED	evidence.based	Interactions in which users explicitly ask about evidence-based biomedical and health sciences research.
		kinesiology	
	tagJUVENILE	juv lit section	Interactions in which users ask about the Juvenile Literature collection or inquire about the availability of children's literature more broadly.
		Juvenile Literature	
		re.escape(juv. lit)	
		children{01}s collection	
		children{01}s lit	
		children{01}s stor	
		re.escape(childrens boooks)	
		(?<!Germany on English)children{01}s boo	
		re.escape(childrens picture)	
		picture book	
	tagCURRICULUM	curriculum materials	Interactions in which users ask about the Curriculum Materials Center or K-12 education materials more broadly.
		curriculum books	
	tagKNOWNITEMARTICLE	doi\W\s{1}\S+	Interactions in which a user identifies a specific, individual journal article.
		doi:{01}\s{01}\d\S+	
		this article	
		this\s\w+\sarticle	
		this paper	
		doi\.\S+	
		doi:{01}\s{01}\d\S+	
		doi\.org\S+	
	tagREFERENCE	articles{01}\sabout	Interactions in which users ask broadly for reference/research support and guidance.
		books{01}\sabout	
		subject	
		topic	
		a paper on	
		help me find an{01}	
	tagURL	re.escape(amazon.com)	Interactions in which a user shares a url to any website.
		re.escape(newfirstsearch)	
		re.escape(galegroup)	
		re.escape(ingentaconnect.com)	
		re.escape(proquest.com)	
		re.escape(ncbi.nlm.nih.gov)	
		re.escape(sciencedirect.com)	
		re.escape(springer.com)	
		re.escape(tandfonline.com)	
		re.escape(webofknowledge)	
		re.escape(wiley.com)	
		re.escape(books.google)	
		re.escape(google.com)	
		re.escape(apps.lib.k-state.edu/databases)	
		re.escape(er.lib.ksu.edu)	
		re.escape(er.lib.k-state.edu)	
		re.escape(getit.lib.ksu.edu)	
		re.escape(getit.lib.k-state.edu)	
		re.escape(guides.lib.ksu.edu)	
		re.escape(guides.lib.k-state.edu)	
		re.escape(catalog.lib.ksu.edu)	
		re.escape(catalog2.lib.ksu.edu)	
		re.escape(catalog.lib.k-state.edu)	
		re.escape(catalog2.lib.k-state.edu)	
		re.escape(primo.hosted.exlibrisgroup.com	
		re.escape(na02.alma.exlibrisgroup)	
		re.escape(searchit.lib.ksu.edu)	
		re.escape(searchit.lib.k-state.edu)	
		re.escape(lib.k-state.edu)	
		re.escape(lib.k-state.edu)	
		re.escape(doi.org)	
		re.escape(http)	
		re.escape(www.)	

APPENDIX IV – EXPERIMENTAL DESIGN, RESULTS, ANALYSES

https://arch.library.northwestern.edu/concern/parent/xg94hp819/file_sets/kk91fk733