An Evaluation of Two Methods for Generating Synthetic HL7 Segments Reflecting Real-World Health Information Exchange Transactions

Thomas S. Mwogi, MBChB MMed[1,2], Paul G. Biondich, MD MS[1,2], Shaun J. Grannis, MD MS[1,2]
[1]Regenstrief Institute, Indianapolis, IN
[2]Indiana University Purdue University (IUPUI), Indianapolis, IN

Abstract

Motivated by the need for readily available data for testing an open-source health information exchange platform, we developed and evaluated two methods for generating synthetic messages. The methods used HL7 version 2 messages obtained from the Indiana Network for Patient Care. Data from both methods were analyzed to assess how effectively the output reflected original 'real-world' data. The Markov Chain method (MCM) used an algorithm based on transitional probability matrix while the Music Box model (MBM) randomly selected messages of particular trigger type from the original data to generate new messages. The MBM was faster, generated shorter messages and exhibited less variation in message length. The MCM required more computational power, generated longer messages with more message length variability. Both methods exhibited adequate coverage, producing a high proportion of messages consistent with original messages. Both methods yielded similar rates of valid messages.

Introduction

Health information is gathered within and among different organizations and technical systems, stored in different formats with different identifiers. This situation requires technology that integrates disparate data, and such technology must be validated through testing. To validate software's ability to meet industry standards including interoperability with pre-existing software, the use of real world data during the testing phase is necessary.

Access to real world data for testing software is however very challenging for various reasons.[1] First, technology developers often lack rights to access identifiable protected health information (PHI). Second, when such access is permitted, strict regulations governing the use of PHI may impose de-identification and data management burdens limiting the efficiency and effectiveness of the testing process. Further, when representative data is unavailable, software developers may substitute simplistic test data that lacks the vagaries, complexities and idiosyncrasies of real-world data. Subsequently, the lack of access to representative transactional testing data may hinder software development and testing, potentially impacting software quality.

The work in this paper was motivated by a real world problem. OpenHIE is a global, open-source collaborative initiative emerging to assist in the strengthening of national health information exchanges for underserved and resource poor settings.[2] It combines several open source components necessary to accommodate large volumes of health information exchange transactions. To test OpenHIE's integrative functionality, we require representative HL7 messages from 'real-world' settings that must be made available to the multiple open source development communities that comprise OpenHIE.

Although general approaches for generating synthetic data have been published.[3-5] Specific methods for generating large volumes of representative synthetic messages approximating real world HL7 transactions are currently not well-described, nor are we aware of any freely available tools to support this need. The HL7 Messaging Workbench tool allows an array of functionality most prominently allowing creation of conformance profiles and standards conformant message segments based on HL7 version 2.[6] However, functionality to insert synthetic content into HL7 version 2 messages is pending.

Consequently, we sought to evaluate the feasibility, efficiency and effectiveness of two methods that generate synthetic HL7 messages using de-identified HL7 data from the Indiana Network for Patient Care, the nation's largest and longest running health information exchange. We chose these two methods based on their ability to generate new data from the original messages.

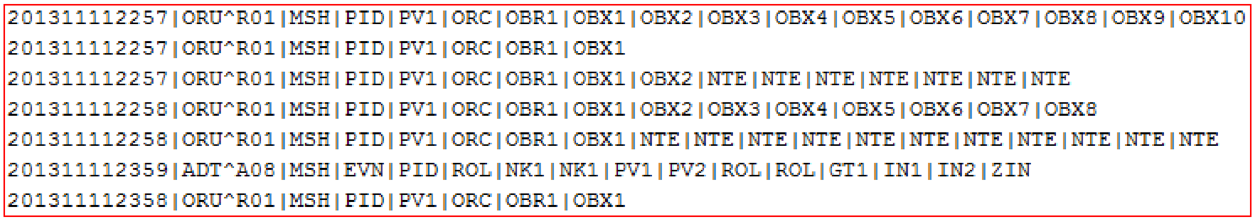
Our process for creating synthetic messages involved specific phases. First, we create a message segment framework consisting of valid HL7 message segments. Once the message segment framework is created, we then inject instance data (simulated patient data, clinical data, etc.) into the appropriate fields in the message. We focus our analysis in this paper on the first phase, message segment creation. We compare the two methods using various metrics, in order to determine the most effective method for generating synthetic HL7 message segment frameworks that accurately approximate the original real-world messages. The comparison metrics included computational effort, level of compliance with the HL7 messaging standard, variability of messages generated, and conformance to the original ‘real-world’ data.

Segment generation in this paper is focused on HL7 version 2. This was largely dictated by the fact that both the HIE source transaction data (from the INPC) is HL7 version 2, and the target software to be tested (OpenHIE) consumes HL7 version 2 messages. HL7 version 2 is one of the most broadly used messaging standards across the world, and in developing countries where OpenHIE is likely to have widespread use, HL7 version 2 offers the advantage of small data footprint, which is well adapted to slow network infrastructures. For this reason, HL7 version 2 has a big role to play and will likely co-exist with other newer standards like HL7 version 3 and Clinical Document Architecture.

Methods

Source Data

We extracted HL7 messages received by the Indiana Network for Patient Care (INPC) during a continuous 24-hour period. The messages were de-identified and stripped down to include only HL7 segments together with the timestamp, source facility, and the message header components including the message type and event type. The figure below illustrates the example source HL7 message segment framework data from the INPC.



```
201311112257|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1|OBX2|OBX3|OBX4|OBX5|OBX6|OBX7|OBX8|OBX9|OBX10
201311112257|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1
201311112257|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1|OBX2|NTE|NTE|NTE|NTE|NTE|NTE|NTE
201311112258|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1|OBX2|OBX3|OBX4|OBX5|OBX6|OBX7|OBX8
201311112258|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1|NTE|NTE|NTE|NTE|NTE|NTE|NTE|NTE|NTE|NTE|NTE
201311112359|ADT^A08|MSH|EVN|PID|ROL|NK1|NK1|PV1|PV2|ROL|ROL|GT1|IN1|IN2|ZIN
201311112358|ORU^R01|MSH|PID|PV1|ORC|OBR1|OBX1
```

Figure 1: Example HL7 message segments extracted from source messages exchanged within the INPC.

We developed a python script to analyze the proportion of HL7 event types in the dataset. The event-type proportions were used to ensure a similar proportion of messages in the synthetic output. We implemted two methods for synthesizing new HL7 message segment frameworks, labelled the Markov Chain Model[7] (MCM) and the Music Box Model (MBM), respectively.

Two Approaches to HL7 message synthesis

Markov Chain Model: A python script was written based on the principles of the markov chain model[7], which is an approach that models nodes (states within a system) and transition probabilities among nodes. In our HL7 segment framework generating process, nodes represent HL7 segments and the transition probabilities represent the probability of transitioniing from one HL7 segment to the next. We developed a transition matrix using the original HL7 source messages. The transition matrix defined the probability of each segment transitioning to the next segment. Transition matrices were generated for each specific HL7 message event type. We then used the transition matrix to generate random HL7 message frameworks that approximated the original messages.

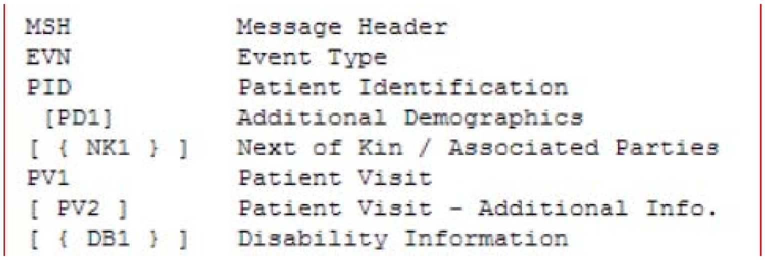
Music Box Model: This model’s name was derived from the music box, a 19th/20th century music instrument that produces sound using a set of pins placed on a revolving cylinder so as to pluck the tuned teeth in what would seem like random plucking movement. For this method we generated HL7 message segment frameworks by choosing messages from the original de-identified source data using simple random sampling with replacement and then adding the chosen message segment to a new generated pool. Using the precalculated proportions for each event type HL7 messages matching the event type were picked at random from the original messages until a desired number of messages were generated.

The two models were evaluated using several parameters to determine the most suitable model that can be used to generate data that were consistent with messages from INPC. These were the parameters of interest:

1. Time required to generate message segment frameworks.
2. Number of segments generated per message.

3. Proporation of message segment frameworks that conform to the HL7 messaging standard rules.
4. Proportion of message segment frameworks that match original messages.

Determining conformance with HL7 standard rules: In order to validate the segment transitions that were generated, we developed a collection of valid segment transitions based on the HL7 message specification. Each HL7 message type contains an abstract message with a collection of segments including rules describing features such as optionality and repetition. Below is an example of an incomplete ADT^A01 abstract message with its rules.



```
MSH            Message Header
EVN            Event Type
PID            Patient Identification
 [PD1]         Additional Demographics
[ { NK1 } ]    Next of Kin / Associated Parties
PV1            Patient Visit
[ PV2 ]        Patient Visit - Additional Info.
[ { DB1 } ]    Disability Information
```

Figure 2: ADT^A01 abstract message with rules

We used a python script to generate all possible valid segment transitions from the information contained in the abstract messages of each trigger event. The above ADT^A01 abstract message would generate the following valid segment transitions separated by commas and so on.

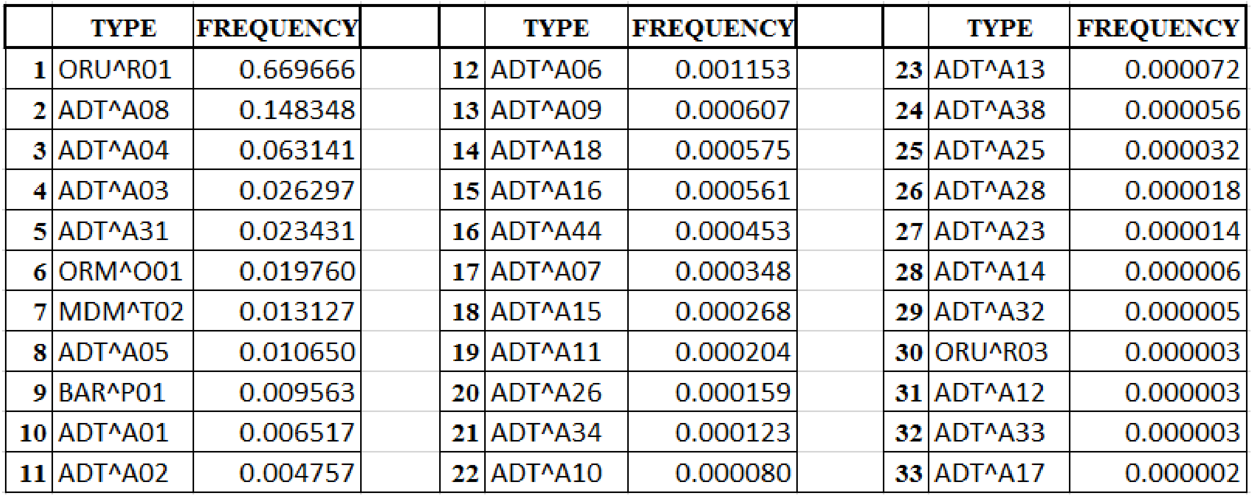
```
MSH|EVN,EVN|PID,PID|PD1,PID|NK1,PID|PV1,PD1|NK1,PD1|PV1,NK1|PV1,NK1|NK1,PV1|P
V2,PV1|DB1,PV2|DB1,DB1|DB1…
```

Using the set of valid segment transitions, we could then determine the percentage of valid segment transitions generated by each of the two models.

Results

Our original INPC data source contained 627,329 representative HL7 messages. The proportion of each message type stratified by trigger event is shown below.

Table 1: Original HL7 message proportions per event type



	TYPE	FREQUENCY		TYPE	FREQUENCY		TYPE	FREQUENCY
1	ORU^R01	0.669666	12	ADT^A06	0.001153	23	ADT^A13	0.000072
2	ADT^A08	0.148348	13	ADT^A09	0.000607	24	ADT^A38	0.000056
3	ADT^A04	0.063141	14	ADT^A18	0.000575	25	ADT^A25	0.000032
4	ADT^A03	0.026297	15	ADT^A16	0.000561	26	ADT^A28	0.000018
5	ADT^A31	0.023431	16	ADT^A44	0.000453	27	ADT^A23	0.000014
6	ORM^O01	0.019760	17	ADT^A07	0.000348	28	ADT^A14	0.000006
7	MDM^T02	0.013127	18	ADT^A15	0.000268	29	ADT^A32	0.000005
8	ADT^A05	0.010650	19	ADT^A11	0.000204	30	ORU^R03	0.000003
9	BAR^P01	0.009563	20	ADT^A26	0.000159	31	ADT^A12	0.000003
10	ADT^A01	0.006517	21	ADT^A34	0.000123	32	ADT^A33	0.000003
11	ADT^A02	0.004757	22	ADT^A10	0.000080	33	ADT^A17	0.000002

The most frequent message triggers in the source data set were ORU^R01, ADT^A08, ADT^A04, ADT^A03 and ADT^A31, which comprised 93.1% of the total messages. These proportions were maintained when generating message segment frameworks.

The two models were used to generate increasingly larger number of HL7 message segment frameworks – from 100 to 1,000,000. The two models were compared with respect to the amount of time required to generate messages, total segments generated and the number and percentage of valid segment transitions in all the messages generated. Table 2 below shows the results.

Table 2: Markov Chain vs Music Box model in generating sequential increasing HL7 messages

	Markov Chain Model vs Music Box Model in generating sequentially increasing messages					
	Markov Chain Model			**Music Box Model**		
Number of HL7 Messages	**Time (sec)**	**Total Segments**	**Valid Segments (%)**	**Time (sec)**	**Total Segments**	**Valid Segments (%)**
100	0	784	741 **(94.5)**	9	742	718 **(96.7)**
250	1	2,470	2,397 **(97.0)**	8	1,681	1,628 **(96.9)**
500	2	4,893	4,762 **(97.3)**	8	3,982	3,842 **(96.5)**
750	3	8,233	7,956 **(96.6)**	8	5,723	5,538 **(96.8)**
1,000	4	9,791	9,443 **(96.5)**	9	7,755	7,504 **(96.8)**
2,500	10	27,547	26,619 **(96.6)**	9	20,449	19,890 **(97.3)**
5,000	17	46,271	44,465 **(96.1)**	10	38,403	37,259 **(97.0)**
7,500	28	74,587	71,999 **(96.5)**	11	54,993	53,300 **(96.9)**
10,000	35	94,401	90,825 **(96.2)**	12	76,626	74,332 **(97.0)**
25,000	93	247,556	238,643 **(96.4)**	18	191,678	185,853 **(97.0)**
50,000	177	481,597	463,539 **(96.3)**	27	375,491	363,900 **(97.0)**
75,000	271	719,318	692,848 **(96.3)**	42	585,645	568,228 **(97.0)**
100,000	355	956,191	920,368 **(96.3)**	48	766,797	743,159 **(97.0)**
250,000	880	2,414,645	2,325,899 **(96.3)**	123	1,933,244	1,874,977 **(97.0)**
500,000	1769	4,860,340	4,682,388 **(96.3)**	222	3,834,268	3,717,745 **(97.0)**
750,000	2752	7,246,021	6,978,775 **(96.3)**	323	5,739,783	5,565,519 **(97.0)**
1,000,000	3695	9,656,978	9,301,474 **(96.3)**	425	7,655,476	7,423,464 **(97.0)**

On average the Markov chain model generated more segments per message when compared with the Music box model. The percent of valid HL7 segment transitions in both models were comparable ranging between 94% and 98%. In both models the proportion of valid message segments generated was independent of the number of messages generated.

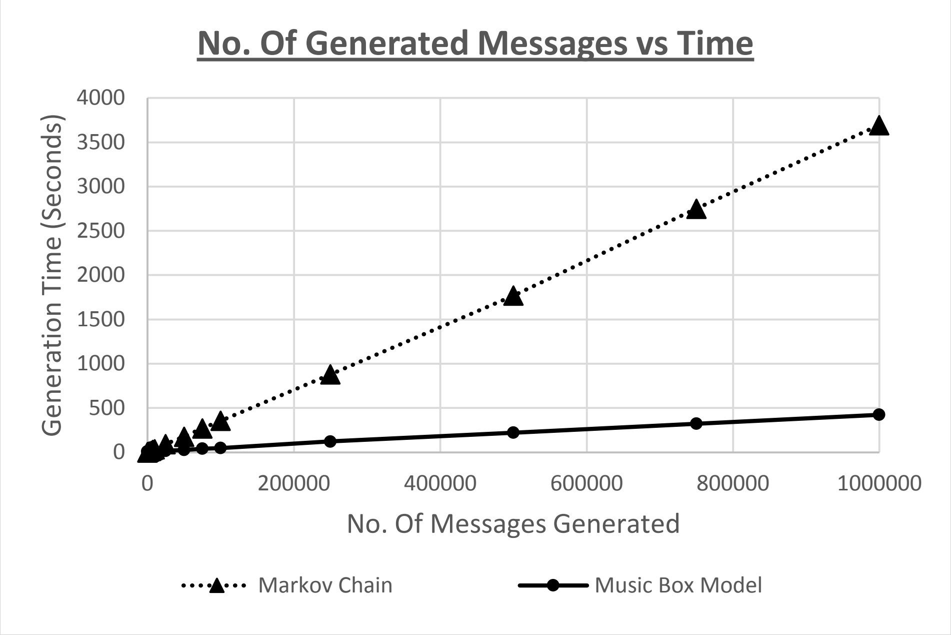


Figure 3: Comparisons of time taken to generate increasing no. of messages by the two models

Because the Music Box method must load all source messages into memory before generating new messages, the Markov Chain method was faster when generating a small numbers of messages (< 2,500). Overall, the music box method was faster. The Markov Chain method generated 270 message segment frameworks per second and the Music Box method generated 1,100 message segment frameworks per second. The message generating rates were nearly constant, as shown in Figure 3.

Comparing coverage of the two models: We assessed the degree to which each method generated message segment frameworks corresponding to one or more of the original source messages, a measure we defined as "coverage". Table 3 below shows the data obtained when the two models generated sequentially increasing number of messages.

Table 3: Comparisons of the two models in terms of coverage of original messages

	Markov Chain Model vs Music Box Model Coverage Of Original HL7 Messages	
	Markov Chain Model	**Music Box Model**
Number of HL7 Messages	**No. of messages identical to original HL7 messages (%)**	**No. of messages identical to original HL7 messages (%)**
100	281,630 **(44.8)**	350,997 **(55.9)**
250	312,527 **(49.8)**	388,518 **(61.9)**
500	356,187 **(56.8)**	418,368 **(66.7)**
750	361,454 **(57.6)**	431,500 **(68.8)**
1,000	396,876 **(63.3)**	456,437 **(72.8)**
2,500	427,513 **(68.2)**	502,382 **(80.1)**
5,000	453,976 **(72.4)**	519,882 **(82.9)**
7,500	463,736 **(73.9)**	527,091 **(84.0)**
10,000	470,447 **(75.0)**	549,014 **(87.5)**
25,000	496,201 **(79.1)**	570,735 **(91.0)**
50,000	506,989 **(80.8)**	584,566 **(93.2)**
75,000	522,169 **(83.2)**	590,466 **(94.1)**
100,000	530,474 **(84.6)**	595,614 **(94.9)**
250,000	540,282 **(86.1)**	606,252 **(96.6)**
500,000	553,554 **(88.2)**	612,750 **(97.7)**
750,000	560,124 **(89.3)**	615,214 **(98.1)**
1,000,000	566,515 **(90.3)**	616,733 **(98.3)**

Overall the Music Box method exhibited greater coverage than the Markov chain method. Both models showed that with an increasing number of HL7 message segment frameworks generated, there was a corresponding increase in coverage of the original messages. The rate of coverage increase plateaued after 75,000 messages. The graph below reflects the data above. Note that the initial 100 messages in both methods corresponded with a more than 40% coverage:

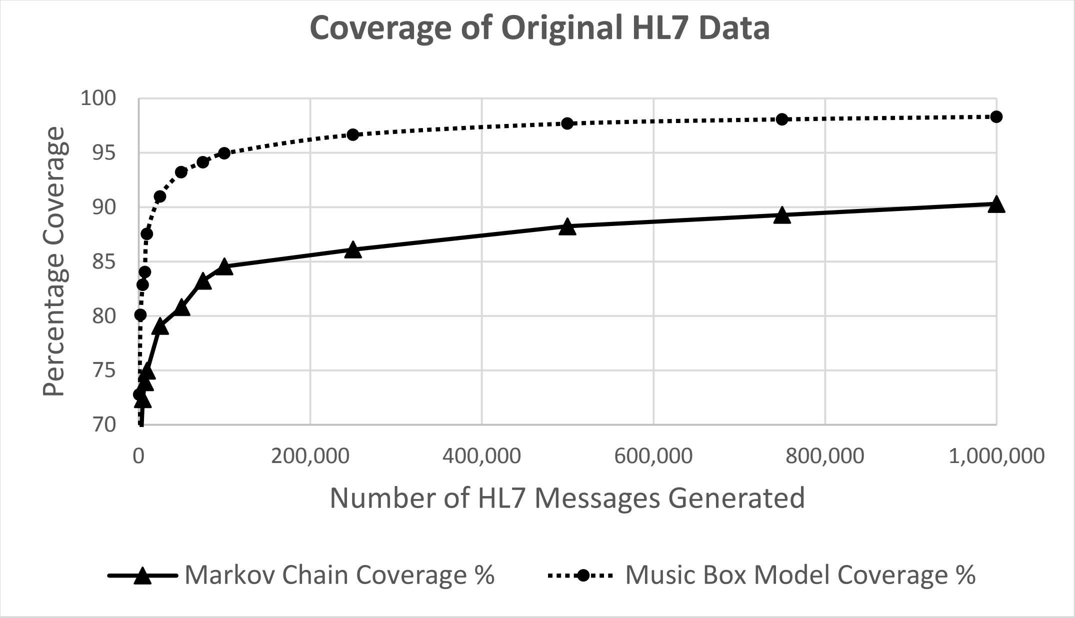


Figure 4: Graph of percentage coverage of original HL7 messages with increasing number of messages generated

The Music Box model had a higher coverage than the Markov Chain for every level of any number of HL7 messages generated. The Music Box model increased to 98.3% compared with Markov Chain at 90.3% when both generated 1 million messages.

Shown below is the comparisons based on segment length per message:

Table 4: Analysis of the number of segments per HL7 message of the two models

	Original Data	Music Box Model	Markov Chain Model
No. of HL7 Messages	627,329	750,000	750,000
Mean	15.67	12.65	14.66
Median	10	10	10
Mode	9	9	9
Std. Deviation	28.24	17.91	34.92
Skewness	14.62	21.32	26.39
Std. Error of Skewness	.003	.003	.003
Minimum	6	6	6
Maximum	1,145	980	3,612

We compared the average number of segments per message in the original 627,329 source messages to 750,000 messages generated by the Markov Chain model and the Music Box model. The most common number of segments per message (mode) generated was 9 in both the original data and both methods. All the data in the three arms were positively skewed with a median of 10 segments. Mean segment length was 15.67 in the original data closely matched by the Markov Chain model at 14.66 segments per message. The spread of segment length per message was also closely correlated between the original data and the Markov Chain as shown by the standard deviation in comparison to that of the Music Box mode. However, the Markov Chain model was prone to generating messages with significantly large number of segments (up to 3,612) compared to the largest number of segments in the original data (1,145).

Discussion

Overall the Music Box model required less time to generate messages when compared with the Markov Chain model. The Music Box model was 5 times faster than the Markov Chain model in generating the same number of messages. This is primarily explained by the fact that the Markov Chain model uses a complex algorithm based on transitional matrix probabilities, which require more computational effort. The Music Box model, on the other hand, implements a straight forward simple random sample of messages, and is therefore preferable when messages must be generated with maximum speed.

Both models generated messages with similar rates of valid segments as gaged by compliance with the HL7 standard rules for each message trigger type. Both topped 97% which reflected to proportion of valid message segments in the Indiana Health Information Exchange source. This rate of valid messages was independent of the number of messages to be generated.

There was a high proportion of identical message segment frameworks in the original dataset. This was reflected by the high coverage by the generated messages. As few as 100 messages generated by both models corresponded to approximately 300,000 (50%) of the original messages. Overall, the Music Box model generated a higher proportion of message segment frameworks corresponding original messages. This is explained by the fact that this method was randomly selecting from the pool of original messages. The Markov Chain method attained a coverage of 90.3% after generating a million messages.

The Markov Chain model generated relatively longer messages when compared to both the original messages and the Music Box model. The Markov Chain also generated greater variability with respect to the number of segments per message compared to the Music Box model. In most situations where software is being tested, this wider variability of messages may be desirable.

Alternatively, the Music Box model generated relatively shorter messages compared with the original messages. This is not necessarily a reflection of the weakness of this method but more a reflection of selecting the most common messages out of the original data. The messages with the largest number of segments were much less common and therefore were less likely to be selected by the Music Box model. However, since these messages are used in generating the transition matrix, the Markov Chain model did reproduce these messages.

Using the mean and standard deviation, the Markov Chain model appeared to more closely correlate with the original data than the Music Box Model. The pattern of the length of messages generated by both models was skewed to the right. Both models showed a statistically significant standard error of skew at 0.03. This means most messages were short with a median segment length of 10. However, messages of up to 3,000 segments per message existed on the Markov Chain model compared to the maximum segment length of 1,145 in the original data. This may have contributed to the Markov Chain method's higher mean and larger standard deviation that correlated more closely with the original data. It is possible that if we constrained number of segments allowed by the Markov Chain method, and thus minimized its outliers, the MCM may have less closely correlated with the original data.

Either method can be used to generate HL7 segments in other settings tailored to the local context. In this paper, we used source data from the Indiana Network for Patient Care. The source data was in the form of de-identified HL7 data stripped down to segment level. Informaticians interested in generating segments using either method in their local context can user our software (made available upon request) to analyze the proportion of HL7 message types in their dataset and then generate transition matrices for each message type. Synthetic message segments for each message type can then be generated according to the transition matrices. For the MBM method, new segments can be generated by randomly selecting each message type in keeping to the analyzed proportions.

There are limitations in making conclusions about HL7 message generation from this study. The synthetic process in this paper focused on generation of HL7 message segment frameworks only and doesn't involve generation of individual field-level data within the segments. Our future work includes developing pragmatic methods for populating fields in each segment with synthetic data that reflect underlying characteristics of the original data. The two methods used to generate message segments can be similarly applied to generate data for individual fields in each segment. The MBM method is being used to randomly select simulated patient identifiers to complete the HL7 patient ID (PID) segment. The MCM will be used to generate simulated address data by first generating transition matrices based on real-world address data. Therefore, both methods are useful not only for generating the segments but also for generating the entire HL7 message.

The MCM’s use of transitional matrices introduces additional complexities beyond simple random sampling that require more computational cycles regardless of the language of code or optimization level, thus we believe that the fundamentally different characteristics of the two approaches accounted for much of the difference observe time to completion. However implementation-specific software inefficiencies also likely contributed to a portion the observed time differences. Thus, different implementations of these approaches may yield different computational efficiency results.

Conclusion

In summary, both methods represent effective approaches to generating message segment frameworks, which are necessary precursors to creating fully realized synthetic HL7 messages. The Music Box model was faster, generated shorter message segment frameworks and had less variability in message segment framework length. The Markov Chain required more computational power, generated longer message segment frameworks with some outliers and had more variability in message segment framework length. Both models demonstrated adequate coverage, generating message segment frameworks corresponding to a high proportion of the original messages. The data generated by both models also had a high compliance with the HL7 standard rules.

The work in this paper forms an important first phase in evaluating important models that can be used to generate HL7 messages that reflect real-world data.

References

1. McHale JV. Using anonymized NHS data without consent: a step too far? British Journal of Nursing. 2012;21(1):54-5.
2. Grannis S, Biondich P, editors. OpenHIE: Helping underserved environments better Leverage their electronic health information through standardization. Proceedings of the 2014 PHI Conference; Apr 29-May 1, 2014; Atlanta, GA.
3. Barse EL, Kvarnstrom H, Jonsson E, editors. Synthesizing test data for fraud detection systems. Computer Security Applications Conference, 2003 Proceedings 19th Annual; 2003 8-12 Dec. 2003.
4. Lin P, editor Development of a synthetic data set generator for building and testing information discovery systems. Proc 3rd Int'l Conf Information Technology; 2006: IEEE CS Press.
5. Houkjaer K, Torp K, Wind R, editors. Simple and Realistic Data Generation. Proc 32nd Very Large Databases; 2006: VLDB Endowment.
6. Workbench M. Developed by Peter Rontey at the US Veterans Administration (VA) in conjunction with the HL7 Conformance Special Interest Group.
7. Markov AA. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. Dynamic Probabilistic Systems. 1971;1(Markov Chains).