Streaming medical report analytics at increasingly "Big Data" scale

Anthony Nguyen, Derek Ireland, Guido Zuccon, Deanne Vickers, Bevan Koopman, Michael Lawley
Australian e-Health Research Centre, CSIRO

1. Summary

A system for high throughput, parallel, text analytics of medical reports is described. The advent and adoption of electronic health records (EHR) has resulted in an exponential increase in the volume of clinical text. Natural language processing (NLP) systems can extract and analyse data from the records for clinical decision support, evidence based medicine and disease surveillance. However, a number of factors limit the successful impact of clinical NLP systems, namely, the ability to process large amounts of clinical data, intrinsic complex nature of narrative medical reports, and the often high computational complexity of NLP systems which inhibits the real-time processing of data streams. In a case study involving the analysis of over 45 million HL7 pathology messages, a messaging service relying on multiple instances of an NLP system executing in parallel was implemented; implementations using 3 instances of the NLP system in parallel achieved a throughput 2.5 times faster than the sequential processing of reports.

2. Introduction

There has been a growing interest in using EHR systems to improve the quality of health care through decision support, evidence based medicine and disease surveillance. Medical records such as discharge summaries, pathology and radiology reports contain a lot of potentially valuable medical information. Unfortunately this information can be difficult to extract and analyse because it is buried in text that may be unstructured, ungrammatical or fragmented. The large amount of textual clinical data that constitutes the majority of the patient record has to be consolidated in order to exploit the information contained in EHRs.

NLP technologies such as data mining and retrieval, machine learning, and information extraction are key to unlocking information in medical records. The computational complexity of NLP systems is often high and the sequential processing of clinical texts has proven adequate only on a small scale. Current approaches, however, are not feasible when working on a larger scale where there is a need to analyse "bigger" data. As a result, efficient and effective techniques to overcome these barriers are of importance for continuing impact.

Due to the little or no dependencies between medical reports, their analysis can be parallelised for reducing the computational time of NLP systems. Messaging services have been proposed to implement a method of communication between the input and output of a medical NLP system to accommodate for real-time data streams or large datasets, as well as to take advantage of the parallelism of NLP processes to enhance efficiency.

3. Description

The XYZ system is a Java-based NLP software platform created at the ABC for the development of clinical language engineering analysis engines to support data-driven analytic tasks [1]. XYZ incorporates clinical domain knowledge through the use of SNOMED CT for unifying the language of the reports for automatic medical text inference and reasoning. It has successfully been used for medical text analytic services on pathology and radiology reports as well as death certificates. XYZ has been applied to small scale datasets of up to 5000 reports for research purposes; however its utility on real-time data streams and larger datasets may be inadequate if the computational time for the analysis of reports cannot keep up with the demands of the incoming data stream.

The Java messaging service (JMS) was chosen as the messaging broker for providing an intermediary to allow Java applications to be loosely coupled and reliably create, send, receive and read messages [2]. This messaging service is built on the concept of message queues, producers (senders), and consumers (receivers). A message producer is used for sending messages to a specific queue. The message consumer is then used for receiving messages from a specified queue. Multiple message consumers can be set up in parallel to receive messages from the same queue such that only one message is only received by one of the consumers. Furthermore, consumers acting on data can publish their results to another queue called a message topic, whereby other consumers wishing to register and subscribe to the topic can receive messages from the topic. This scenario allows for multiple consumer applications to act on the same messages published from a given consumer.

Pathology report case study

The proposed medical text analytic messaging service for analysing HL7 messages from a State-wide pathology information system to perform Cancer Registry tasks such as the notification of cancer reports and the coding of notifications data is illustrated in Figure 1. The pathology data was obtained with research ethics approval from the YYY. Apache ActiveMQ[1], an open source message broker, which fully implements JMS, was used to realise the messaging service. The message producer (HL7 Producer) accesses pathology HL7 messages and through the selection of report types that are relevant for subsequent processing, messages were sent to a specified queue (REPORTS_QUEUE). Multiple XYZ consumers can be set-up such that each consumer will take a message from the queue in turn. The results from the XYZ analysis are encoded in JSON[2] format and published to a message topic (RESULTS_SUBSCRIPTION) where the topics can be subscribed to by end-user applications (Results Consumer), for example, to consolidate patient results and store them in a database and/or provide support for clinical coders to abstract clinical information from medical reports.

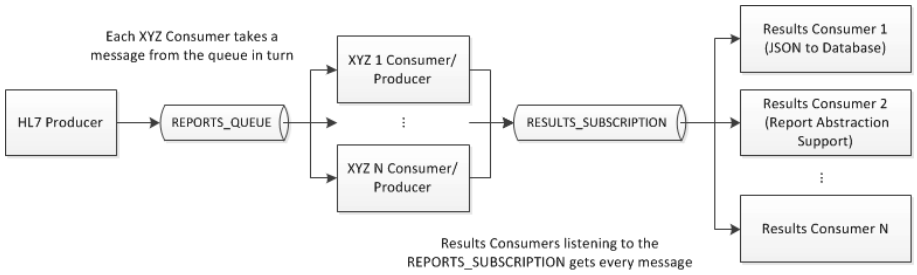

Figure 1. Messaging architecture for analysing HL7 messages.

4. Results

The messaging service has been applied to a State-wide pathology HL7 message feed. HL7 messages from the year 2009 were used to analyse the pathology reports as well as test the load on the service. In total, 45,307,035 messages were available, from which 119,581 were considered relevant for subsequent processing by XYZ. Using 3 instances of XYZ, the system's average processing rate was 3.6 seconds per message and achieved the batch processing of all the messages within just under 5 days.

Results have shown an increase in report analysis throughput by using the messaging framework and multiple instances of XYZ consumers in parallel. The use of 3 XYZ instances in parallel resulted in a 2.5 times speed-up over the sequential single instance of XYZ in operation. Dependent on system resources, further speed-ups are possible if additional instances of XYZ and/or multiple instances of XYZ's shared resources such as the SNOMED CT ontology and concept mapping servers were made available.

5. Conclusion

Analysis of the contents of EHRs will have a profound impact on clinical care. The use of the messaging technologies that take advantage of the parallelism of consumers/producers can be an effective real-time processing solution for data streams. These technologies greatly increase the throughput of medical NLP analytics for clinical decision support and/or research activities involving real-time data streams or large datasets.

[1] First Author*, et al.*, "Title," *Journal Name,* vol., pp., Month, Year.
[2] M. R. Richards*, et al.*, *Java Message Service*, 2nd ed. ed. Sebastopol, CA: O'Reilly, 2009.

[1] http://activemq.apache.org/
[2] http://www.json.org/