MSDS 7330 term project: Post-Market surveillance of Illumina products

Ivan Charkin (icharkin@smu.com), Jeff Leath (jleath@smu.edu), and Alec Neff (alecn@smu.edu)

***Abstract*—Illumina is a leading company in the DNA sequencing market. It produces several platforms of DNA sequencers for various applications and consumables used for sequencing runs. Since the company's products are used in medical applications, such as prenatal screening and cancer research, there is a need to actively monitor their performance in the hands of the customers. One way to proactively detect problems with products is to monitor user chat rooms. In this project, we analyzed data from Twitter and created a demonstrative database for the information contained within the Twitter comments. The comments were preprocessed using natural language processing and then organized for easy search.**

I. Introduction

Illumina is currently offering a variety of products on the market for various applications of DNA sequencing [1]. It sells several DNA sequencers platforms with various capabilities for both research and medical applications. The company also produces consumables (such as flowcells) and reagents which are used by customers for DNA sequencing runs. There exist many regulations for medical application products which require proactive collection of data related to the product performance and use after the product is launched [2]. Although planned proactive data collection methods such as customer surveys and experts user groups may identify product performance issues, these methods may be too slow to identify problems affecting customers in the field. For example, if there is a new recurring problem or inconvenience stemming from a recent software update in the field, the problem may not be flagged by Illumina until the next survey or, if the issue is serious enough, through the recorded customer complaints. There is a need to create a tool allowing immediate response to any potential problems with the product once it is in the hands of customers.

II. Methodology

A potentially faster method of proactively identifying problems with the product is monitoring the user discussion and chat groups. Multiple websites exist wherein users of Illumina products discuss potential problems with those products and seek solutions (for example, www.seqanswers.com).

MySQL was used for demonstrating a schema of a relational database. MySQL is the world's most popular open source database. It is available both in a free of charge version or enterprise edition available to businesses[3]. For this project we used MySQL Community Server release 8.0.13. MySQL Workbench was used to generate the schema.

Twitter is an American online news and social networking service on which users post and interact with messages known as "tweets". Tweets were originally restricted to 140 characters, but on November 7, 2017, this limit was doubled for all languages except Chinese, Japanese, and Korean. Registered users can post, like, and retweet tweets, but unregistered users can only read them. Users access Twitter through its website interface, through Short Message Service (SMS) or its mobile-device application software ("app") [4], [5]. Due to the number of character limits and the information on users provided by the platform, it is ideal for testing the data collection for this project.

In order to have access to Twitter data programmatically, we created an app that interacts with the Twitter application programming interface (API). There are many APIs on the Twitter platform that software developers can engage with, with the ultimate possibility to create fully automated systems which will interact with Twitter [6]. For this project we chose the oldest package created for this purpose: Tweepy. It is an easy-to-use Python library for accessing the Twitter API which has extensive documentation and use demonstrations online [7].

MongoDB was used to store the information obtained from Twitter. MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) [8].The benefit of MongoDB is that it stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time [9].

Textblob package was used to do the text processing. TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [10]. The textblob.sentiments module contains two sentiment analysis implementations: PatternAnalyzer (based on the pattern library) and NaiveBayesAnalyzer (an NLTK classifier trained on a movie reviews corpus).

III. Results

At the start of the project we created a mock relational database using MySQL. We assumed collection of comments from multiple websites. A post will have an author we likely can identify and track. The same post may mention multiple products made by Illumina. The posts may further be shared by other other users. Technically the same user may be posting on multiple websites, but since aliases are likely to be used, it

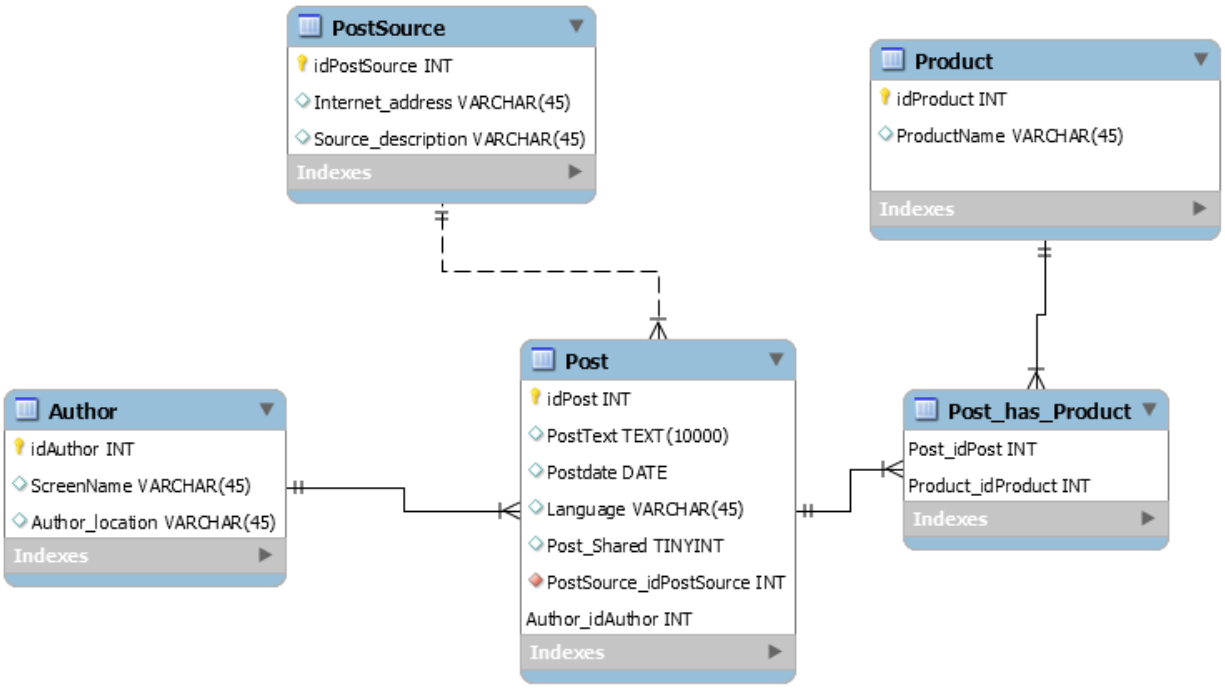

Fig. 1. Schema of a relational database with the posts obtained from multiple sources.

is unlikely we can track users over multiple websites. The schema is shown in Figure 1. We can continue expanding the database schema, however, it is unlikely to provide any useful insight into the data contained in the text of the post. We would likely have to perform preprocessing of the text prior to logging the record into the database and add attributes based on the Natural Language processing algorithms. Due to the complexity of connecting to various websites and creating customized scrapers, we did not drive this solution to the full completion in this project.

Instead of trying to collect information from multiple websites, we chose instead to concentrate on Twitter to test the data collection and rudimentary NLP processing. We created a Python script utilizing Tweepy package which searched though tweets on Twitter for the following keywords: ”Illumina”, ”MiSeq”, ”MiniSeq”, ”NextSeq”, ”NovaSeq”, ”iSeq”, ”HiSeq”. By doing this we collected tweets generally discussing Illumina as well as mentioning its DNA sequencing platforms currently available on the market.

Tweepy returned the following parameters in the Status object: *__class__, __delattr__, __dict__, __dir__, __doc__, __eq__, __format__, __ge__, __getattribute__, __getstate__, __gt__, __hash__, __init__, __init_subclass__, __le__, __lt__, __module__, __ne__, __new__, __reduce__, __reduce_ex__, __repr__, __setattr__, __sizeof__, __str__, __subclasshook__, __weakref__, _api, _json, author, contributors, coordinates, created_at, destroy, display_text_range, entities, favorite, favorite_count, favorited, full_text, geo, id, id_str, in_reply_to_screen_name, in_reply_to_status_id, in_reply_to_status_id_str, in_reply_to_user_id, in_reply_to_user_id_str, is_quote_status, lang, metadata, parse, parse_list, place, retweet, retweet_count, retweeted, retweets, source, source_url, truncated, user*.

Note that to obtain non-truncated text of the tweets we had to change the default setting of the Tweepy query. If one wants to get full-text on Twitter response, it is necessary to add a keyword tweet_mode=’extended’ when calling API. By adding this keyword, obtained full_text field in the response from the API, instead of text field.

The Status object of tweepy itself is not JSON serializable, but it has a _json property which contains JSON serializable response data as a dictionary. By using json.dumps(tweet._json) command, we extracted all relevant information about every single tweet in JSON format. We saved it on a hard drive for later upload to a database.

When running the query it was obvious that many of the returned tweets with the same text were repeated multiple time due to retweeting. A Retweet is a re-posting of a Tweet. Twitter’s Retweet feature helps you and others quickly share that Tweet with all of your followers [5]. Sometimes people type ”RT” at the beginning of a Tweet to indicate that they are re-posting someone else’s content. This isn’t an official Twitter command or feature, but signifies that they are quoting another person’s Tweet. Although knowing when specific tweet is being retweeted multiple times is useful for certain applications, we modified the query to eliminate retweets from the results. First we eliminated the tweets starting with ”RT” by just checking a tweet’s text to see if it starts with ’RT’. Then we eliminated all tweets marked as ”retweet”.

To analyze the collected data we loaded the information into MongoDB by creating ”Illimuna” database with a collection called twitter. To populate the database we used ”mongoimport”. The ”mongoimport” tool imports content

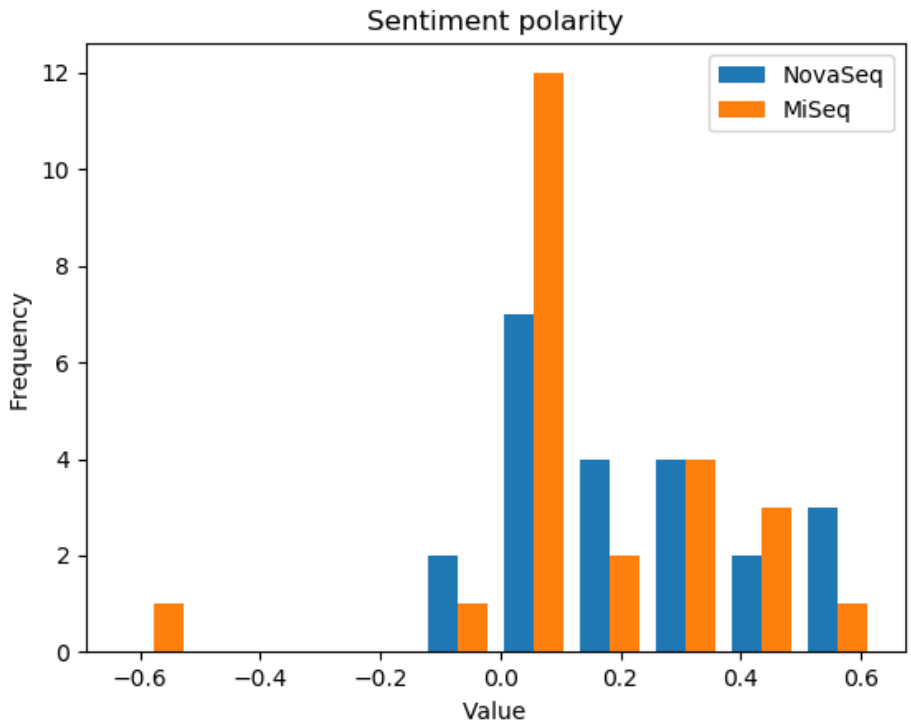

Fig. 2. Sentiment alalysis for the selected Illumina products.

from an Extended JSON, CSV, or TSV export created by mongoexport, or potentially, another third-party export tool [9]. Since Twitter allows collecting tweets only for the last 7 days for free accounts, we had to collect the data several time. For subsequent updates of the database we used ”upsert” option of the mongoimport. By default, mongoimport uses the _id field of JSON file to match documents in the collection with documents in the import file. To specify that to match existing documents for the upsert mode, we used –upsertFields with the ”id” of the twitter.

For this project we did not do any preprocessing of the text before placing it in the database. Since the dataset is small we can do text analysis on the text queried from the database. For larger dataset a carefully designed preprocessing of the text is needed to avoid long wait time.

To test the database we created a query in Python which searched through a ”full_text” field for specific keywords, such as product name, and returned only full text of tweets which fit the search criteria. For example when searching for tweets with ”novaseq”, it will return all tweets which have the word ”novaseq” in it. We used ”$regex” operator which provides regular expression capabilities for pattern matching strings in queries. Option ”i” makes the query case insensitive.

One of things which are obvious when we look at the tweets is that they are written in different languages. Therefore, to obtain useful information on all tweets, we utilized a TextBlob ”translate” function which automatically detects the source language and translates to the specified language [10]. Language translation and detection is powered by the Google Translate API. Although the automatic translation may not be perfect, it can provide us with adequate information for further analysis.

As an example of the data which can be obtained from text analysis we calculated Sentiment of every tweet. Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral. Sentiment property of TextBlob returns a namedtuple of the form Sentiment(polarity, subjectivity). Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Figure 2 shows the result of the sentiment analysis returned for NovaSeq and MiSeq products. One can see that in the time frame of the analyzed tweets the majority of the tweets were positive. One can see that there is an outlier for sentiment with the polarity equaling -0.64. We identified the tweet which generated the low polarity score. It stated ”*Just shy of 4 TeraBases for our S4 NovaSeq run. That is seriously mad!*” This is obviously not a negative statement as the author is likely expressing amazement at the amount of genetic data his new Illumina instrument is producing. The negative value of the polarity is likely cause by words ”shy” and ”mad”. Further human analysis or machine learning supervised training will be needed to obtain more accurate NLP generated data.

IV. Conclusions

We obtained Illumina users’ comments from Twitter and organized them in a database. The database can be queried by product allowing for the analysis of consumer sentiment to

allow the company to identify potential problems with specific products.

REFERENCES

[1] "Illumina products website," https://www.illumina.com/products.html, accessed: 2019-02-03.

[2] I. Tariah and R. Pine, "White paper: Effective post-market surveillance," BSI Group, Tech. Rep. BSI/UK/440/ST/0614/en/HL. [Online]. Available: https://www.bsigroup.com/meddev/LocalFiles/en-US/Whitepapers/WP-Post-market-surveillance.pdf

[3] "Mysql product datasite," https://www.mysql.com/, accessed: 2019-03-31.

[4] Wikipedia contributors, "Twitter — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Twitter&oldid=889118467, 2019, [Online; accessed 2-April-2019].

[5] "Twitter website," https://www.twitter.com/, accessed: 2019-03-31.

[6] "Accessing the twitter api with python," https://stackabuse.com/accessing-the-twitter-api-with-python/, accessed: 2019-03-31.

[7] "Tweepy website," www.tweepy.org, accessed: 2019-03-31.

[8] Wikipedia contributors, "Mongodb — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=MongoDB&oldid=890993425, 2019, [Online; accessed 7-April-2019].

[9] "Mongodb website," https://www.mongodb.com/, accessed: 2019-03-31.

[10] "Textblob: Simplified text processing," https://textblob.readthedocs.io/en/dev/, accessed: 2019-03-31.