International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# Automated User Experience Research is made possible by Artificial Intelligence

*Rahul Sharma,*

,

*Abstract. This paper describes an approach to a prototypic system that aims to automate user research activities. Building digital products that fulfil user needs requires an analysis of its users, their contexts and their tasks. These user research activities commonly require a large amount of human effort, resulting in collections of unlinked quantitative and qualitative data sets. Our system aims to analyse and combine "human-made" user research data and extract meaningful insights through machine learning methods. This paper highlights the challenges I faced while building the first prototypes, our need for labelled data and how I managed to overcome its scarcity. Finally, I describe the first prototype which consists of a keyword extractor, sentiment extractor and text annotator, providing a programmatic way to analyse user research data.*

Keywords: Machine learning · Natural language processing · Neural networks · User research · Qualitative analysis · Quantitative analysis

## 1 Introduction

In business, most companies regard the customer experience of their digital products or services as the primary differentiator towards competition [1]. Referring to the standard "Ergonomics of human-system interaction – […] Human-centred design for interactive systems" [2], designing a digital customer experience needs to be based on user (ex-perience) research activities and the insights gained. In simple terms, user research is conducted by collecting and analysing quantitative data (e.g. programmatic user tracking, online surveys or customer requests) and/or qualitative data (e.g. contextual inquiries, field studies or focus groups). Gathering this information, especially quan- titative data, may in part be done in an automated fashion. HoIver, combining different data types, analysing that combination and deriving insights from it, requires a very high degree of human effort. In many contexts the way user research is set-up, planned and conducted is highly inefficient, especially in regards to the following three aspects: 1) Qualitative information is usually used for one-off projects which results in high costs for a limited outcome, 2) merging data is a manual process requiring intense human effort, and 3) linking quantitative and qualitative data is rarely done.

## 2 Objective

Our objective is to develop a system that automates user research activities. This system consolidates and links quantitative and qualitative user research data, analyses it and creates meaningful user data insights. With the help of this system, companies save time and money in manual analysis and uncover hidden potential in their own data to optimize or innovate research and development topics, business models and customer experiences. This paper describes the approach towards an initial prototype of that system and indicates how I gained first insights.

## 3 Approach

Before starting to work on the first prototype, I outlined the main challenges for user researchers when it comes to understanding users. I wanted to determine how a programmatic approach, especially machine learning1 (ML), may be effectively applied to meet researchers' needs. It is not a simple task to analyse a huge amount of unstructured contextual user documentation in order to obtain deep insights from it. With machine learning and natural language processing2 (NLP) techniques, investi- gating the user context becomes faster and easier.

To create an initial prototype, I Int through three phases. First, I used fun- damental NLP methods to implement a keyword extractor tool that enables researchers to highlight the main user concerns in survey data. After that, I implemented a sentiment extractor since it is important for businesses to understand how customers perceive their products. Finally, I built a text annotator to associate meaningful tags with data from user evaluations.

## 4 Statistics Based Insights

With NLP, statistical methods can be used to derive insights from text, such as key- word extraction. Keyword Extraction. Keyword extraction is an automatic process that consists of extracting the most relevant words or expressions from a text. It helps to pull out the most important words or phrases that best describe a user statement on a huge file of statements. It can be achieved using statistical methods, linguistic methods or machine learning algorithms. In our case, I used a popular statistical method called TF-IDF. TF-IDF stands for term frequency-inverse document frequency. It is a mathematical formula that measures how relevant a word is to a given document [3] – in our example

[1] Machine learning is an artificial intelligence (AI) field that enables systems to learn "intelligent" tasks through experience. To build an ML algorithm, I have two fundamental steps: *Training* where the algorithm learns how to perform the desired task and *testing* to validate the model.

[2] Natural language processing is an artificial intelligence component that aims to make computers understand and generate human natural language.

below, a single user statement from a user review – in a collection of many documents. It considers the whole set of reviews when setting the keywords. The TF-IDF method calculates the number of occurrences of a word in a review, term frequency, and compares it with the inverse document frequency which indicates how common this word is in the whole

data set of reviews. The higher the TF-IDF score, the more relevant the term is to the review. Thus, I consider words with the highest TF-IDF values as keywords. In one example, I used previously gathered user requirements from user stories3 as input data. The following shows two examples of keyword extraction using TF-IDF:

```
As a user, I want a clearer grid icon so that I can better understand the connection betIen both screens.
=> Screen, Icon, Connection
As a user, I want to see the product's season so that I can have a better overview.
=> Season, Product, Overview
```

## 5 Machine Learning Based Insights

In this section I describe our journey with machine learning methods, namely neural networks[4], to perform sentiment analysis and text classification for the creation of our prototype.

### 5.1 Data Collection

Data is the lifeblood of every machine learning project. During training, ML models learn how to predict results from data. Thus, the choice of data is critical for the model quality and efficiency. In our project, I needed labelled data to perform sentiment analysis and text classification, since both are supervised learning[5] techniques. Sentiment Dataset. Sentiment analysis is a binary classification problem: The data is labelled into positive or negative statements. Thankfully, there are many free datasets to train sentiment classification models. I chose to work with the IMDB data set which contains 50,000 native movie reviews [4]. The dataset is balanced and split into training and testing sets, each containing 25,000 user reviews. Moreover, each set contains 12,500 positive reviews and 12,500 negative reviews. Text Classification Dataset. Our text classification model is a multi-labelling problem, which means, that for every user statement I associate a set of labels.

[3] A user story in agile software development is an informal, natural language description of one or more features of a software system from an end-user perspective.

[4] Neural networks are a subclass of machine learning algorithms, that are mainly inspired by biological neural networks of mammals.

[5] In supervised learning, I have data and labels (as an output) and I want the algorithm to learn how to map the input data to the desired output.

I prepared our dataset from scratch because it was challenging to find a suitable dataset, especially considering I want a specific set of labels for every data point. For this, I collected customer feedback data from the internet. More specifically, I scraped reviews from Amazon's Ibsite and labelled the data manually.

For every user comment I associated some, all or none of the following tags: Pricing, Ease of Use and Features. Although labelling is crucial for model accuracy, it is costly and time consuming. So far, I only labelled 1,600 reviews and have

man- aged to deal with this lack of data using algorithmic methods.

As a first step, I used data augmentation to expand the size of the labelled set without any human effort. I opted for a synonym replacement method which consists of replacing a noun or a verb with its most similar synonym. For every review, I replaced 50% of its nouns and verbs. The final augmented set consists of 3,000 comments.

## 5.2 Data Processing

Since the collected data is mainly a range of native users' statements, it is unstructured and not in the desired format. Therefore, I cleaned the text to enhance its quality. I also encoded it into numerical vectors using word embeddings so that it is readable by a machine learning algorithm.

Text Cleaning. The goal of text cleaning is to get rid of extra useless information and put the raw sentences in a proper form. The first step is tokenization which consists of splitting the sentences into a group of tokens or words. The next step is normalization, which is a series of operations that puts all user statement texts on the same level. From those operations I mainly used lemmatization [5] to set a word to its lemma or root word, stemming to eliminate affixes, removing punctuation such as commas and quotes, as Ill as removing extra whitespaces. I also set all words to loIrcase and remove stop words, which are common words like "that", "and", "or", with no meaningful semantic contribution.

Word Embedding. Word embedding is a way to map words from a vocabulary to vectors of real numbers. This transformation is necessary because machine learning algorithms such as neural networks are not able to process strings of plain text. This vector representation has two main advantages that make it poIrful above other encoding methods. It guarantees a low dimensional vector space which is more efficient when it comes to model complexity and it preserves semantic meanings when repre- senting words, which means that words with similar meaning have similar vector representation. In our neural networks, I started with a word embedding layer to process the already cleaned user statements.

## 5.3 LSTM Neural Network

To perform sentiment analysis and text annotation, I have chosen to work with the long short-term memory (LSTM) [6] neural network, a specific class of recurrent neural networks (RNN) [7].

RNNs are specific neural networks, popular for processing sequential information. In an RNN, neurons have feedback connections which form a sort of internal "mem- ory". The memory means that the RNN performs the same operation for every element of a sequence of inputs in a way that every output is dependent on all previous computations. In each step the network still remembers information about the previous steps. This information is called hidden state. This mechanism helps learn and model sequential data like text as a sequence of words, or words as a sequence of letters.

A major problem with RNNs is their short-term memory: If a sequence is long, RNNs are not able to carry information from initial steps to later steps. If I process long paragraphs with an RNN, it may ignore important information from the beginning and, as a result, deliver bad results. LSTM was invented to overcome this gap by adding another internal mechanism called gates that can regulate the flow of infor- mation through steps.

## 5.4 Sentiment Analysis

The goal of sentiment analysis is to classify native users' statements into two cate- gories: Positive and negative. To train and test our LSTM neural network, I used the labelled reviews from the IMDB dataset, as described in Sect. 5.1. The final model is trained during 4 epochs6 and has a training accuracy of 92% and testing accuracy of 88%.

Below are examples of user statement sentiment classification results with the degree of confidence of each prediction.

```
I don't find the system stable.
=> Sentiment: negative | Prediction score: 0.30341896 Great quality!!
=> Sentiment: positive | Prediction score: 0.57020533
```

## 5.5 Text Classification

The goal of text classification is to assign a set of predefined tags to a given text. It is similar to a keyword assignment task. It is not necessary that the tags appear in the original text. Rather, they help categorize the text content. From a machine learning perspective this is a multi-label classification task, which means that each sample may be assigned to one or many classes at once. This is different from multi-class tasks such as sentiment analysis, in which each sample is only assigned to one.

To train the LSTM model I used our prepared data, as described in Sect. 5.1, which contains 3,000 multi-tagged user statements. I took out 500 samples for later testing. Therefore, I only had 2,500 statements to train the model and here lies the real challenge.

On the one hand, 2,500 labelled samples are not enough for a machine learning supervised model to perform Ill. On the other hand, I have thousands of unlabelled user statements. Unfortunately, I cannot proceed to unsupervised learning7 since I know our target classes. Therefore, I decided to proceed with semi-supervised learning, which combines both supervised and unsupervised learning and makes use of the abandoned unlabelled data.

The semi-supervised technique that I used is called pseudo labelling [8]. It fol- lows four basic steps: First, I train the model on a small amount of labelled data, then I use that first model to predict labels for new unlabelled data. From the new pseudo- labelled set, I select a set of data that corresponds to a good prediction score and combine it with the original

training set, and I continue training. I repeat those steps until I have a good model.

In our case, I started by the 2,500 labelled review that I have, training the model during equal number of epochs, before enhancing the training set. Namely, after each 50 epochs, incorporating the pseudo-labelled data and continuing training. The total number of training iterations was 500, when the model training accuracy stabilized at around 98%. During the final 100 epochs, the model was trained on almost 20,000 multi-labelled statements, and I got a testing accuracy of around 87%.

The following shows an example of a multi-label classification.

```
Predicted tags for 'This is a nice and cheap android device. HD display and sturdy built quality with 10 h + battery backup. Sometimes runs a little slow, rest is good':
Pricing : 95%
Ease of Use: 98%
Features: 99%
```

## 6 Summary, Limitations and Perspectives

With our prototype I have successfully implemented a keyword extractor, sentiment extractor and a text annotator. This is only a first step. With the insights that I have derived so far, a user researcher is able to dig into the voluminous information gathered from and about users more easily. While keyword extraction and sentiment analysis modules can be used to investigate any kind of user statements, the current text classification module is limited to user reviews related to a specific category of products. This is due to the data I used to train the algorithm.

Data is a challenge for us since it determines the quality and the performance of the model. I used user input data such as online user reviews, rather than contextual user research documentation because it was easier to access larger numbers of the former. HoIver, the data sets used for the prototype are not large enough to train the actual system in the required way. I are aware that our first prototypes are not fully accurate, but I are working to get better results and bypass major problems I encountered during training, most notably the text classification model where I had an overfitting[8] problem. For that I are still improving our text classifier and playing with data quality.

As a next step, I are planning to mix insights together, such as merging the text classifier and the sentiment analysis to derive more detailed information. I are also considering building models to extract the users' intent and predict their behaviour. These in-depth insights are necessary to attain our objective of developing a system that automates user research activities.

References

1. Gartner: Customer Experience Research. USA (2017)
2. International Organization for Standardization: DIN EN ISO 9241-210 Human-centred design for interactive systems (2010)
3. Leskovec, J., Rajaraman A., Ullman J. D.: Mining of Massive Datasets, pp. 8–9. Cambridge University Press, Cambridge (2014)
4. Lakshmipathi, N.: IMDB Dataset of 50 K Movie Reviews (2019) . https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. Retrieved 31 January 2020
5. Torres-Moreno, J.: Automatic Text Summarization (Cognitive Science and Knowledge Management). p. 24. Wiley-ISTE (2014)
6. Elman, J.L.: Finding structure in time. Cogn. Sci. 14(2), 179–211 (1990)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)
8. 8. Dong-Hyun, L.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: ICML 2013 Workshop: Challenges in Representation Learning (WREPL), Atlanta, USA (2013)