Topic Modelling FDA Recall Text Data

Eric Pettengill
Jeff Church

Abstract

A study by Zhang et al.[11] examined medical device recalls in the United States from the years 2012-2015 with the goal of determining the impact of software user-interface (UI) errors on patient safety. To identify recalls due to software UI errors, a laborious manual classification process was used. To reduce the manual labor involved in this process, we applied Latent Dirichlet Allocation (LDA) to the task of automatically identifying recalls due to software UI errors.

Using LDA, approximately 96% of software UI recalls were successfully identified. However, due to the similarity between recall descriptions there was also a high number of false positives. Overall accuracy could potentially be improved through further data pre-processing. While the task of automatically identifying software UI recalls was not solved completely, LDA successfully reduced the number of recalls that must be manually classified.

Introduction

Natural language processing can be very useful in analyzing and modelling text data. In our project we were tasked with developing a model that successfully predicts user-interface(UI) software errors and software errors as classified manually in Zhang et al.[11] using text data from the FDA. We were given data from years 2012-2015 as well as the labelled results from Zhang et al.[11]. In total, there were 7,771 recalls with 423 due to software UI errors. Each recall was given an FDA determined cause along with a manufacturer recall reason among many other variables. The goal is to use the manufacturer recall reason text data in order to train a NLP model to predict these recalls labelled software UI errors.

Methods

We decided to use a topic modelling approach to solving this problem, and in particular used Latent Dirichlet Allocation(LDA)[3]-more on this later-using the `topicmodels`[5] R package. Prior to fitting the model there was some data pre-processing to take care of. First, we merged the FDA manufacturer recall text data with the labelled results from Zhang et al.[11]. Second, we created a new error label. Each of the 7,771 recalls used in the paper above was labelled as a non-error or any combination of the following three errors: UI-SW(user-interface software error), CTRL-SW(control software error), and SW(software). We merged these three categories into one, that is, if a recall was labelled any of the three we labelled it an error otherwise a non-error. Next, we filtered out recalls based on the FDA determined cause as the paper above. Lastly we removed all stop words and considered all unigrams, bigrams and trigrams in form of a document term matrix. We then trained an LDA model with data from years 2012-2014 and used year 2015's data to assess our model.

Model

LDA is an unsupervised topic modelling algorithm that identifies a given number of topics that best generate a text corpus. A topic is a list of words along with a distribution over those words (i.e. some words will appear more frequently than others within a topic). Training an LDA model on a corpus produces two collections of data: one containing the identified topics, and another containing the mix of topics that comprises each document in the corpus. Table 1 below shows topics identified in the FDA dataset.

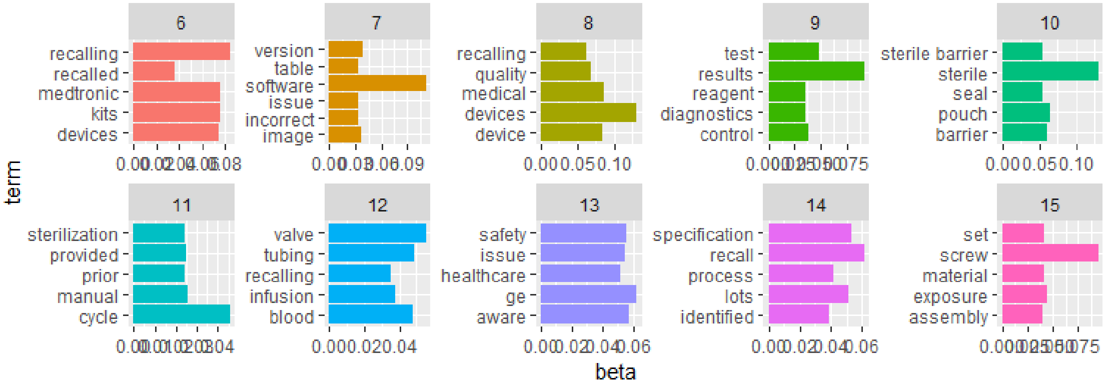
To gain an intuitive understanding of LDA, it's useful to explore the generative process it assumes. LDA assumes that documents are built by randomly drawing the appropriate number of words from the documents' topics. For example, a three-topic, 100-word document comprised of 50% technology, 30% business, and

Table 1: Document/Topic Probabilities

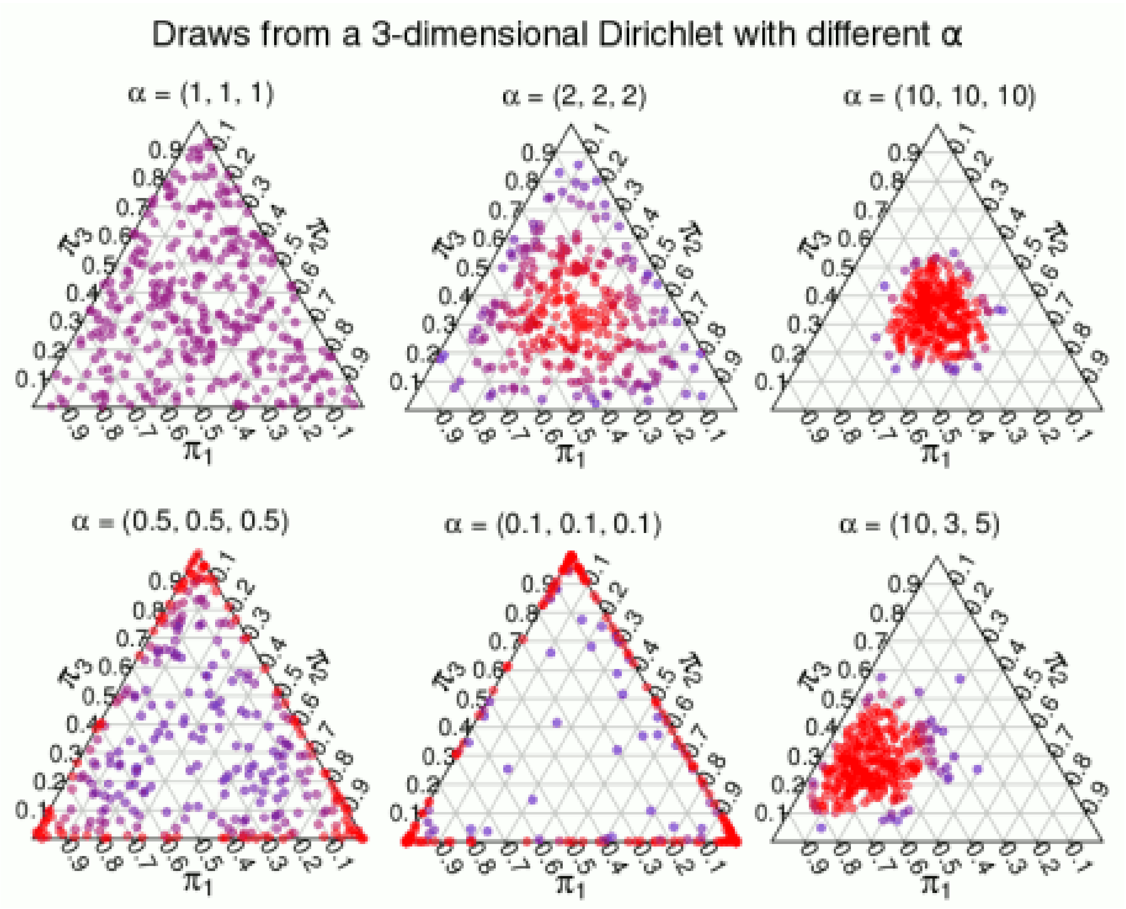
	Predicted Topic Probabilities		Predicted Topic
RECALL_NUMBER	1	2	topic
Z-1214-2015	0.70	0.30	1
Z-2479-2015	0.44	0.56	2
Z-0122-2015	0.36	0.64	2
Z-0190-2015	0.44	0.56	2
Z-2482-2015	0.45	0.55	2
Z-0037-2015	0.56	0.44	1
Z-0146-2015	0.36	0.64	2

20% arts would be made up of 50 words randomly drawn from the technology topic, 30 from the business topic, and 20 from the arts topic. LDA is a bag-of-words model; word order and syntax are not considered. Therefore, any document generated by this process would not be human-readable, but its topics could likely still be discerned. LDA executes this process in reverse; starting with a corpus of documents, the topics and document compositions are determined.

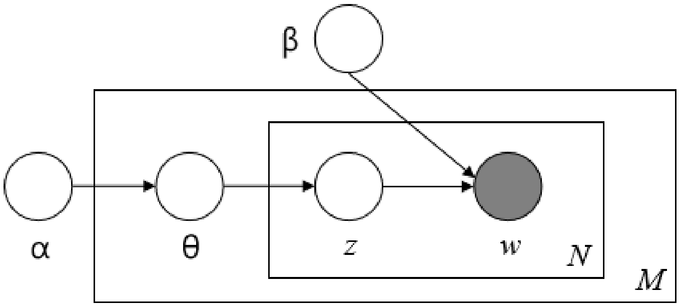
Because LDA is an unsupervised algorithm these topics will not be named, but it's sometimes possible to infer what a topic is from its most common words. For example, in the figure below it's easy to see that topic 10 refers to incidents of sterile conditions being lost due to breaches in packaging. Other topics, for example topic 11, are less obvious.



LDA may be tuned with two parameters; alpha and beta. The value of alpha specifies how similar or dissimilar documents in the corpus are to one another, while beta has a similar effect on the topics. Lower values of alpha and beta indicate less similarity. For the task of classification, low alpha values are desirable because it causes documents to be more heavily represented by a single topic.



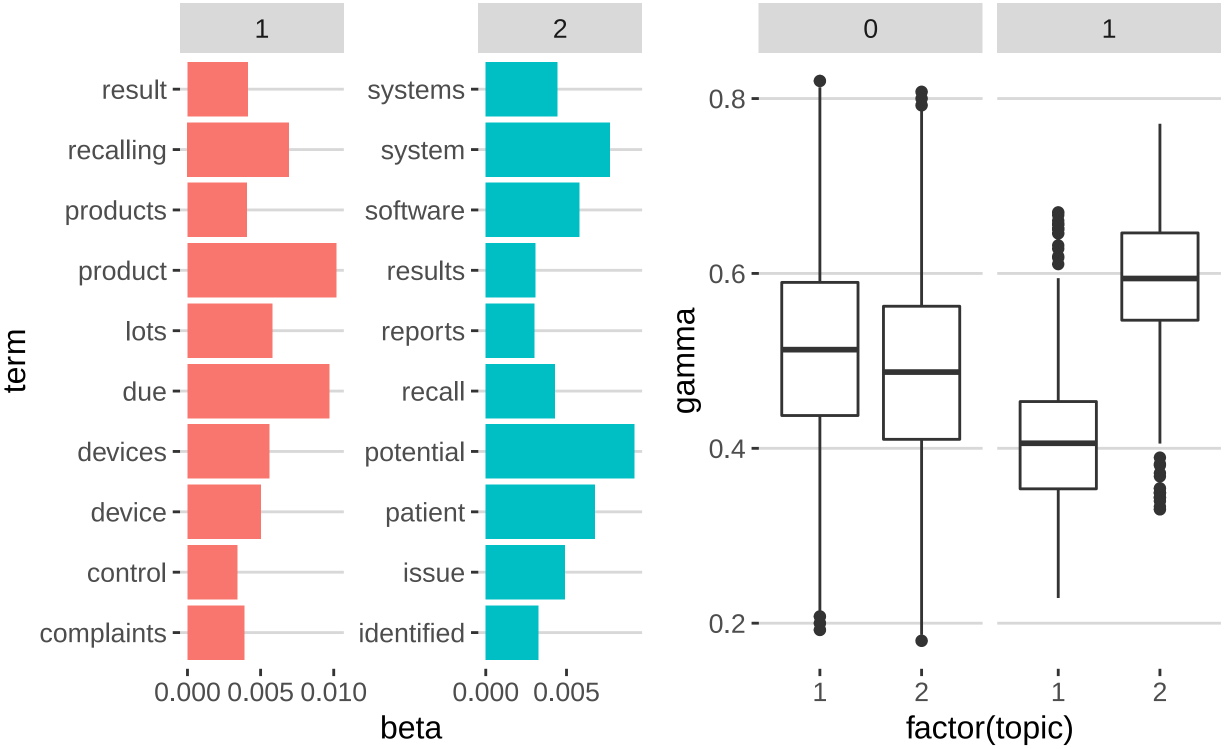
The effect of varying the alpha parameter is shown in the figure above[2]. Topics are represented by the corners of the triangle. Lower alpha values drive the documents to a single topic or a mix of two, while higher values produce a more even mix of all three topics.



Above is a graphical model representation of LDA. M represents documents, and N represents the assignment of topics (z) to words (w). Theta represents the topic distribution of the document, which is influenced by the alpha parameter.

Results

Using the LDA model with 2 topics we trained using data from years 2012-2015. An advantage of using LDA is that it's a probabilistic method. That is, we define the underlying data belongs to 2 topics and the model calculates the probability that each recall belongs to one of the two topics. Along with these probabilities, it also calculates the probability that each word(or bigram/trigram) belongs to each topic. This is useful because we can visualize the top words belonging to each topic as well as the the distribution of document topic probabilities by their being labelled an error or non-error.



The two plots shown above are the top 10 words that occur in each topic-1 and 2-assigned by the trained LDA model(left) and the distribution of each recall(right) belonging to each topic-1 and 2-split by the recall being labelled an error(1) or non-error(0).

Now, with LDA being an unsupervised method we must interpret the topics given by the LDA model, namely, topics 1 and 2. From the word/topic plot on the left we can see that the model is picking up some interesting words belonging to topic 2, notably, the words systems, system, software, and potential, which corresponds to our task of classifying a recall as an error. However, the word "recalling" holds a high probability of occuring in topic 1 whereas the word "recall" is given a high probability of occuring in topic 2. This can be troublesome due to the two words likely meaning the same thing in the context of labelling a recall as an error or non-error. The document/topic plot on the right also reveals some information in interpreting topics. Most notably, recalls that are labelled errors(1) have a higher probability of occuring in topic 2 of the LDA model, whereas, recalls labelled non-errors(0) have a higher probability of occuring in topic 1. From this we will define topic 2 as a recall being classified an error and topic 1 being classified a non-error.

Validation

Now that we have a model trained and topics defined we can use it to make predictions on our test data from year 2015. The confusion matrix is shown below. There are a total of 1,879 recalls in our test data with 122 having been labelled an error.

Table 2: Confusion Matrix

	No-error	Error
	0	1
1	758	5
2	999	117

* ACC.: 46.5%

† TPR: 95.9%

Overall accuracy of the model(46.5%) is not that great but due to class-imbalance we included the true positive rate as an alternative evaluation metric. The true positive rate is 95.9%, that is, it was able to correctly identify approximately 96% of recalls labelled errors. However, there are 5 false negatives(true errors labelled non-error-topic 1) and 999 false positives(true non-errors labeled error-topic 2). The false negatives(Table 3) and a few false positives(Table 4) are shown below.

Table 3: False Negatives

Topic Probabilities		Predicted	Actual	
1	2	topic	error	MANUFACTURER_RECALL_REASON
0.53	0.47	1	1	When performing calibration, an alert message on the spectral filtration of the X-ray beam may be suppressed. Improper filtration of the X-ray Beam can then occur in exams set up with copper filtration.
0.51	0.49	1	1	Under certain circumstances the patient Demographics in a report exported into the EMR may not match the demographics shown in the corresponding Synapse CV clinical report.
0.52	0.48	1	1	Possibility for system display freeze during CT interventional procedures.
0.51	0.49	1	1	A problem exists in MOSAIQ resulting in the incorrect field size being sent to the treatment machine for stereotactic plans using cones.
0.51	0.49	1	1	Multiple software and hardware issues with device that can affect its function.

Table 4: False Positives

Topic Probabilities		Predicted	Actual	
1	2	topic	error	MANUFACTURER_RECALL_REASON
0.37	0.63	2	0	Beckman Coulter is recalling the Small Form Factor Console PC (SFF PC) (B23083) for use with the Access 2 Immunoassay systems because it may experience a "MFC Exception" error during normal operation of the Access 2 Immunoassay Systems.
0.41	0.59	2	0	Beckman Coulter is recalling the Small Form Factor Console PC (SFF PC) (B23083) for use with the DxC 600i analyzer because it may experience a "MFC Exception" error during normal operation of the analyzer system.
0.50	0.50	2	0	Two lots of the Anti-Lambda APC-H7 antibody are contaminated with CD38 antibody.
0.50	0.50	2	0	There is a low probability the scanner arm will become completely detached from the scanner column.

There are a few things to note from the tables listed above:

1. Reading the text of each recall, it's very difficult to discern some as being labelled an error or not.
2. Most of the incorrectly classified recalls have probabilities of belonging to topic 1 and topic 2 very close to one another. We used the default method for classifying a recall. That is, a recall is classified to the topic having the largest probability of belonging to that topic. This could potentially be modified to eliminate false negatives and improve the true positive rate, assuming an error labelled a non-error is more costly than a non-error labelled an error.
3. There are many words(other than stop words) that seem to have no importance in classifying a recall an error or non-error but adds probability to belonging to topic 1 or 2 by sheer abundance.

Conclusion

Using Latent Dirichlet Allocation we were able to correctly identify ~96% of recalls labelled being an error. However, due to the similarity between error and non-error text data used, overall accuracy was hindered. One solution to this would be to manually set the classification rule based on topic probabilities to maximize the true positive rate whilst limiting the false negative rate. At the very least, this method has the potential to reduce the amount of recalls classified manually significantly and could very well be fine tuned to eliminate nonsignificant words used interchangeably between recalls labelled error and non-error.

Acknowledgements

This project was made exponentially easier thanks to the following packages: `tidyverse`[9], `tidytext`[8], `kableExtra`[12], `broom`[7], `topicmodels`[5], `readxl`[10], `tm`[4], `ggthemes`[1], and `patchwork`[6].

References

[1] Jeffrey B. Arnold. *ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'*. R package version 4.1.0. 2019. URL: https://CRAN.R-project.org/package=ggthemes.

[2] Rasmus Baath. *The Non-parametric Bootstrap as a Bayesian Model*. 2015. URL: https://www.r-bloggers.com/the-non-parametric-bootstrap-as-a-bayesian-model/.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[4] Ingo Feinerer and Kurt Hornik. *tm: Text Mining Package*. R package version 0.7-6. 2018. URL: https://CRAN.R-project.org/package=tm.

[5] Bettina Grün and Kurt Hornik. *topicmodels: Topic Models*. R package version 0.2-8. 2018. URL: https://CRAN.R-project.org/package=topicmodels.

[6] Thomas Lin Pedersen. *patchwork: The Composer of ggplots*. R package version 0.0.1. 2017. URL: https://github.com/thomasp85/patchwork.

[7] David Robinson and Alex Hayes. *broom: Convert Statistical Analysis Objects into Tidy Tibbles*. R package version 0.5.1. 2018. URL: https://CRAN.R-project.org/package=broom.

[8] David Robinson and Julia Silge. *tidytext: Text Mining using 'dplyr', 'ggplot2', and Other Tidy Tools*. R package version 0.2.0. 2018. URL: https://CRAN.R-project.org/package=tidytext.

[9] Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1. 2017. URL: https://CRAN.R-project.org/package=tidyverse.

[10] Hadley Wickham and Jennifer Bryan. *readxl: Read Excel Files*. R package version 1.2.0. 2018. URL: https://CRAN.R-project.org/package=readxl.

[11] Yi Zhang et al. "User Interface Software Errors in Medical Devices: A Study of US Device Recall Data". In: *Biomedical Instrumentation & Technology* (May 2019).

[12] Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*. R package version 1.1.0. 2019. URL: https://CRAN.R-project.org/package=kableExtra.