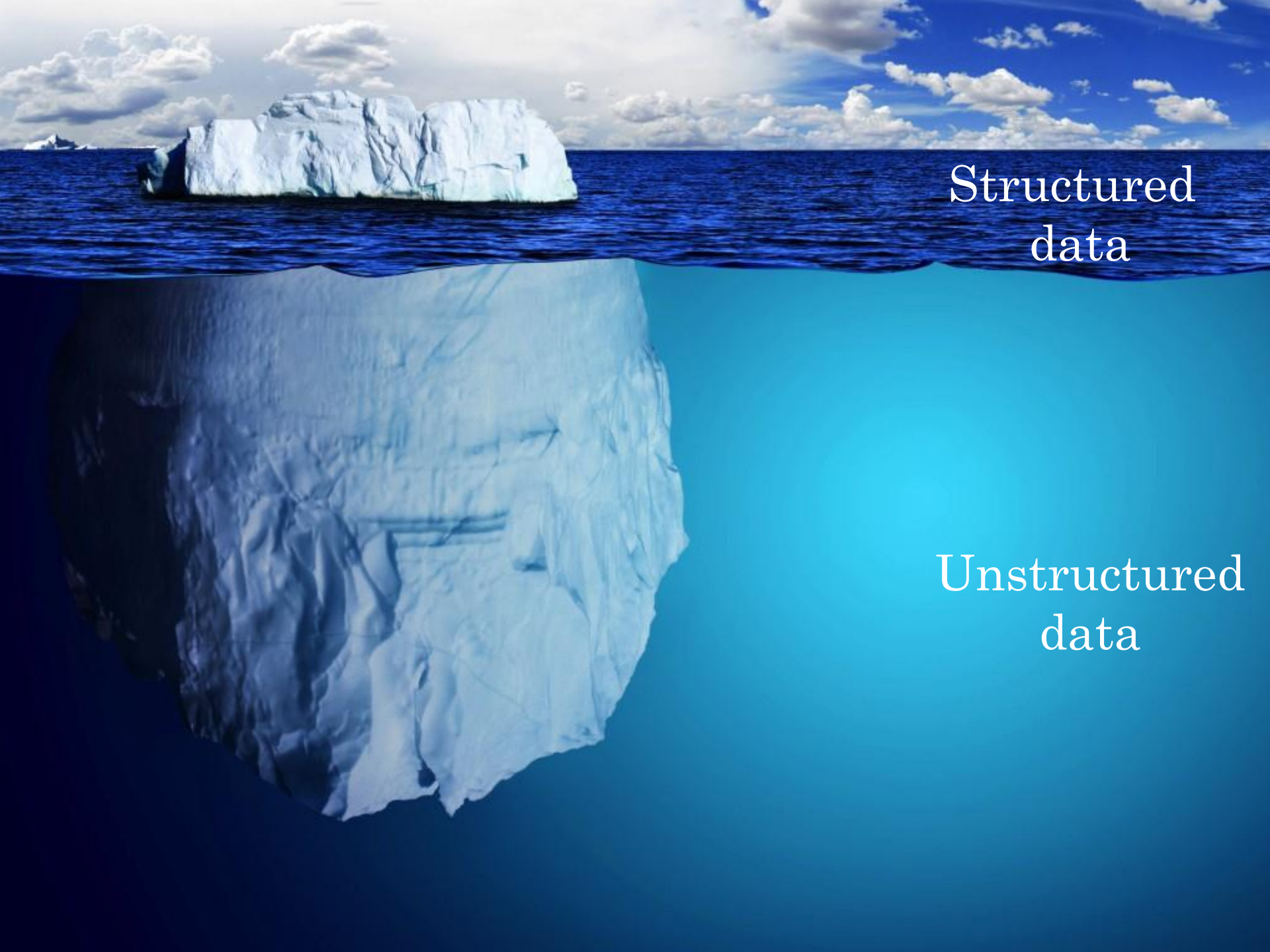
What will happen to this patient? Predictive text mining in a specialist cancer hospital

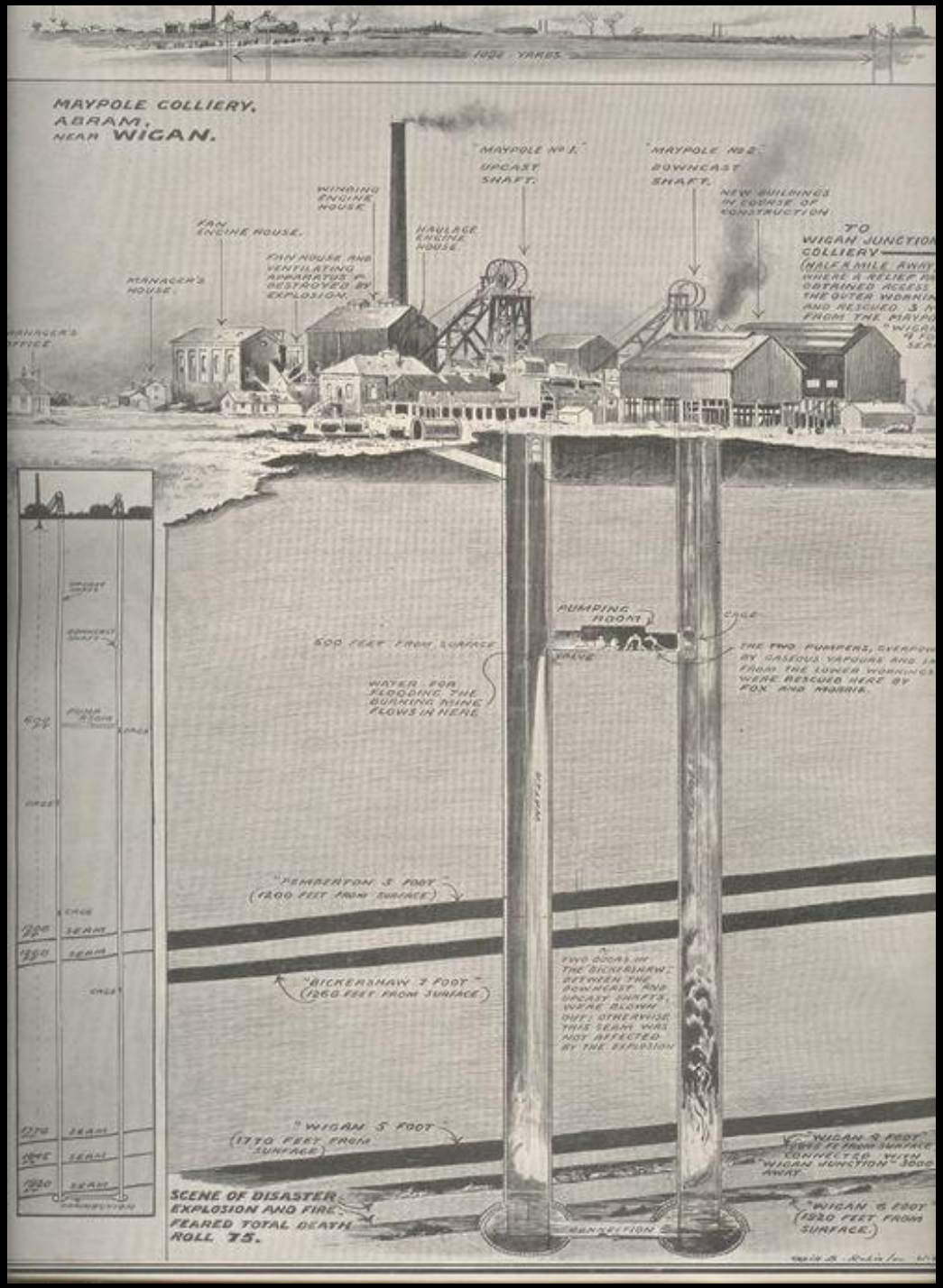
Tom Liptrot
The Christie NHS Foundation Trust, Manchester

EARL 2015

The Christie NHS
NHS Foundation Trust


MANCHESTER
THE CHRISTIE


Structured data
Unstructured data



Structured data

Unstructured data

Prediction using regression

$$Y = \beta X + \varepsilon$$

If Y is	Use
continuous	linear regression
binary	logistic regression
categorical	multinomial logit
time to event	cox model
time series	ARIMA, GARCH
Etc…	

If X is	Use
continuous	use as is / transform
categorical	dummy encoding
is very 'wide'	regularisation
Etc…	

text ??

Prediction using regression

$$Y = \beta X + \varepsilon$$

If Y is	Use
continuous	linear regression
binary	logistic regression
categorical	multinomial logit
time to event	cox model
time series	ARIMA, GARCH
Etc…	

If X is	Use
continuous	use as is / transform
categorical	dummy encoding
is very 'wide'	regularisation
Etc…	

text ??

Text to X – bag of words

1. "The dog barked"
2. "The cat meowed"
3. "The cat and dog ran"

the	dog	barked	cat	meowed	and	ran
1	1	1	0	0	0	0
1	0	0	1	1	0	0
1	1	0	1	0	1	1

Bag of words - problem

1. Cancer was progressing now in remission
2. Cancer was in remission now progressing

cancer	was	progressing	now	in	remission
1	1	1	1	1	1
1	1	1	1	1	1

N-grams

"Cancer was progressing now in remission"

2-grams

Cancer was, was progressing, progressing now, now in, in remission

3-grams

Cancer was progressing, was progressing now, progressing now in, now in remission

Skip N-grams

"The cat and dog ran"

1-skip 2-grams

The * and

Cat * dog

And * ran

2-skip 2-grams

The * * dog

Cat * * ran

Steps in R

1. Import text
2. Clean text
3. Count n-grams
4. Make matrix
5. Fit model
6. Make predictions

Making a term document matrix in R

```
library(tm) #load the tm package
corpus_1 <- Corpus(VectorSource(txt)) #creates a 'corpus' from a character
vector


corpus_1 <- tm_map(corpus_1, content_transformer(tolower))

corpus_1 <- tm_map(corpus_1, removeWords, stopwords("english"))

corpus_1 <- tm_map(corpus_1, removePunctuation)

corpus_1 <- tm_map(corpus_1, stemDocument)

corpus_1 <- tm_map(corpus_1, stripWhitespace)


tdm <- TermDocumentMatrix(corpus_1)


Library(Rweka)
four_gram_tokeniser <- function(x, n) {
        RWeka:::NGramTokenizer(x, RWeka:::Weka_control(min = 1, max = 4))
        }

tdm_4gram <- TermDocumentMatrix(corpus_1,
                control = list(tokenize = four_gram_tokeniser)))
```

The Elastic Net – glmnet package

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$$

```
library(glmnet)
#convert term-document-matrix to a sparseMatrix
X = sparseMatrix(tdm_4gram$i, tdm_4gram$j, x = tdm_4gram$v)

#function fits a one-v-all logistic regression
fit_fun = function(i, y, x){
        #y is a factor variable
        #x is a term-document matrix

        #select one level
        lev = levels(y)[i]

        #make x and y variables
        y_i = y == lev

        glm_i <- cv.glmnet(x = x, y =  y_i,  family= "binomial")

        glm_i
}

out <- alply(1:nlevels(y), 1, fit_fun, x = x, y = y)
```

Automatic tokenisation – textreg package

Fast Logistic Regression for Text Categorization with Variable-Length N-grams

Georgiana Ifrim
Max-Planck Institute for Informatics
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
ifrim@mpi-inf.mpg.de

Gökhan Bakır
Google Switzerland GmbH
Freigutstrasse 12
Zürich, Switzerland
ghb@google.com

Gerhard Weikum
Max-Planck Institute for Informatics
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

A common representation used in text categorization is the bag of words model (aka. unigram model). Learning with this particular representation involves typically some pre-processing, e.g. stopwords-removal, stemming. This results in *one* explicit tokenization of the corpus. In this work, we introduce a logistic regression approach where learning involves automatic tokenization. This allows us to weaken the a-priori required knowledge about the corpus and results in a tokenization with variable-length (word or character) n-grams as basic tokens. We accomplish this by solving logistic regression using gradient ascent in the space of all n-grams. We show that this can be done very efficiently using

1. INTRODUCTION

The standard bag of words representation is widely used in text categorization as an explicit tokenization of the training text, before employing learning algorithms. Typically, some language dependent pre-processing is employed, such as stop words removal or stemming. Furthermore, a feature selection step [32] is often crucial for computational efficiency and generalization. Such feature-engineering often requires detailed knowledge about the language of the text to be categorized. In practice, this results in a lot of tuning of the classifiers in order to find the right unigram features.

However, there are important text classification tasks for which the initial unigram bag-of-words representation does

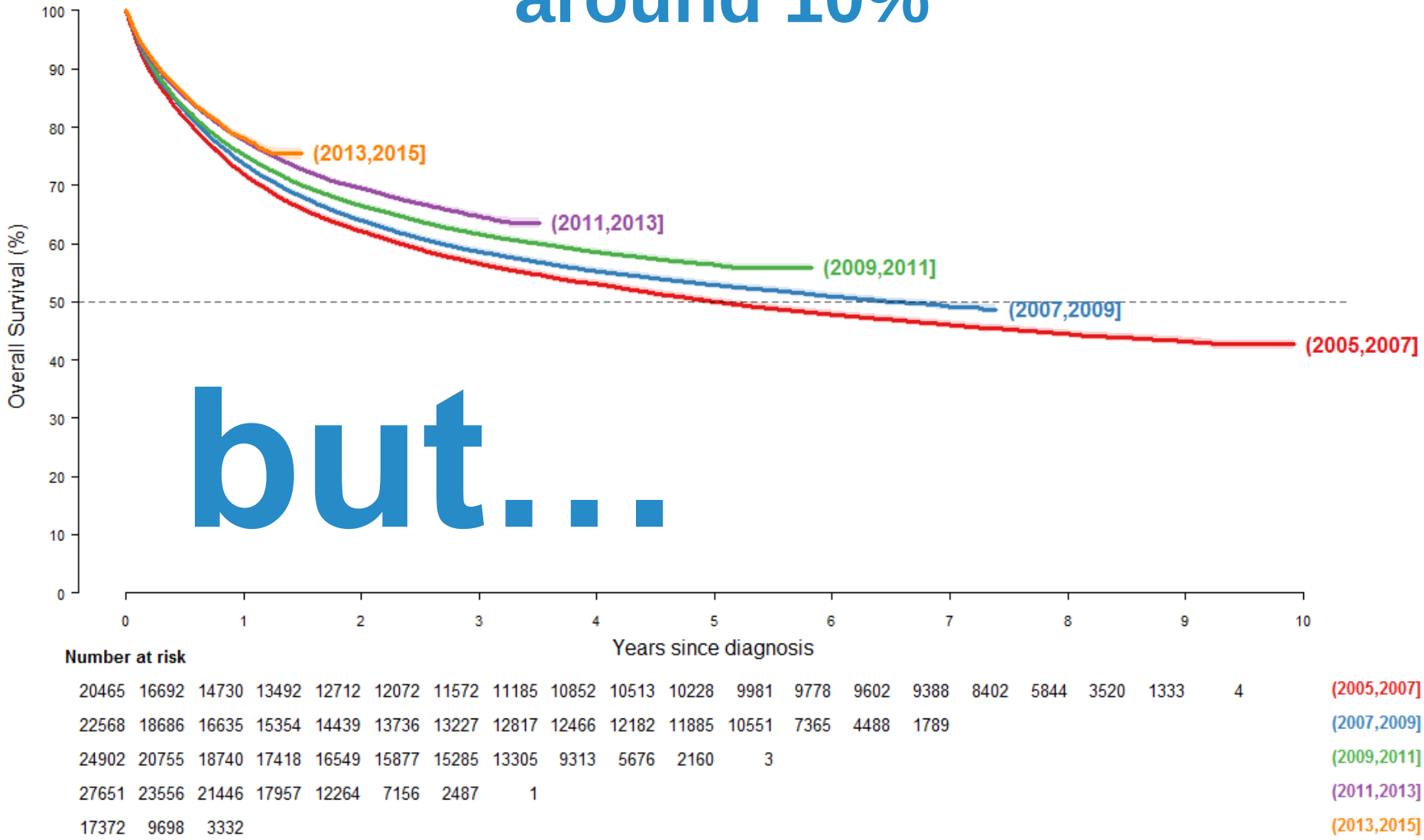
Implementation notes

- You need lots of memory
- I used AWS with 64GB RAM and rewrote much of the tm package
- Store processed text in a local database for repeated analysis
- data.table / dplyr for pre-processing
- Save incremental results/ matrices with saveRDS

EXAMPLE 1:

Mining for structured data

Patient survival has improved by around 10%



but...

Number at risk

20465	16692	14730	13492	12712	12072	11572	11185	10852	10513	10228	9981	9778	9602	9388	8402	5844	3520	1333	4	(2005,2007]
22568	18686	16635	15354	14439	13736	13227	12817	12466	12182	11885	10551	7365	4488	1789						(2007,2009]
24902	20755	18740	17418	16549	15877	15285	13305	9313	5676	2160	3									(2009,2011]
27651	23556	21446	17957	12264	7156	2487	1													(2011,2013]
17372	9698	3332																		(2013,2015]

135,000

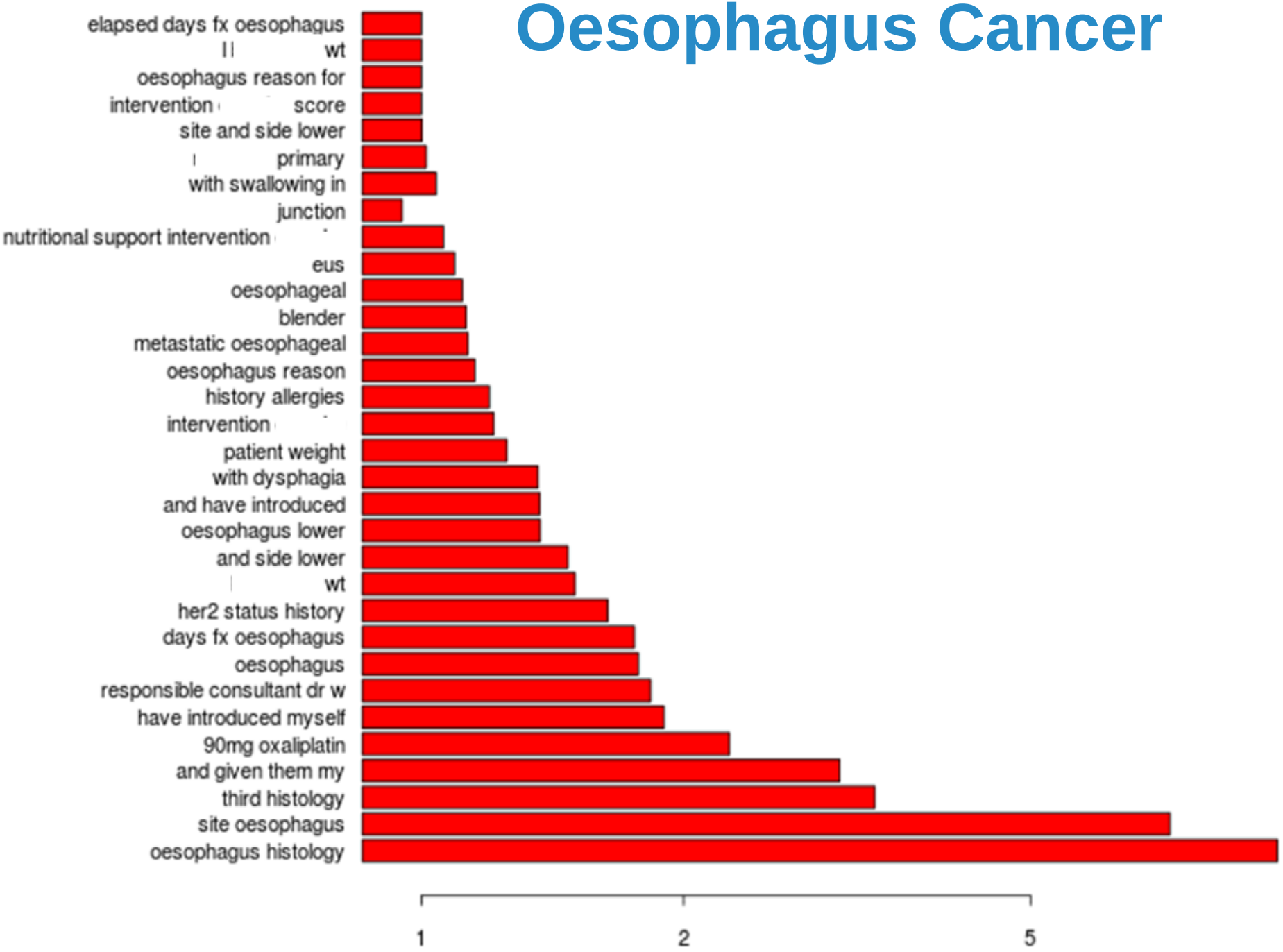
Sets of notes

Would take around

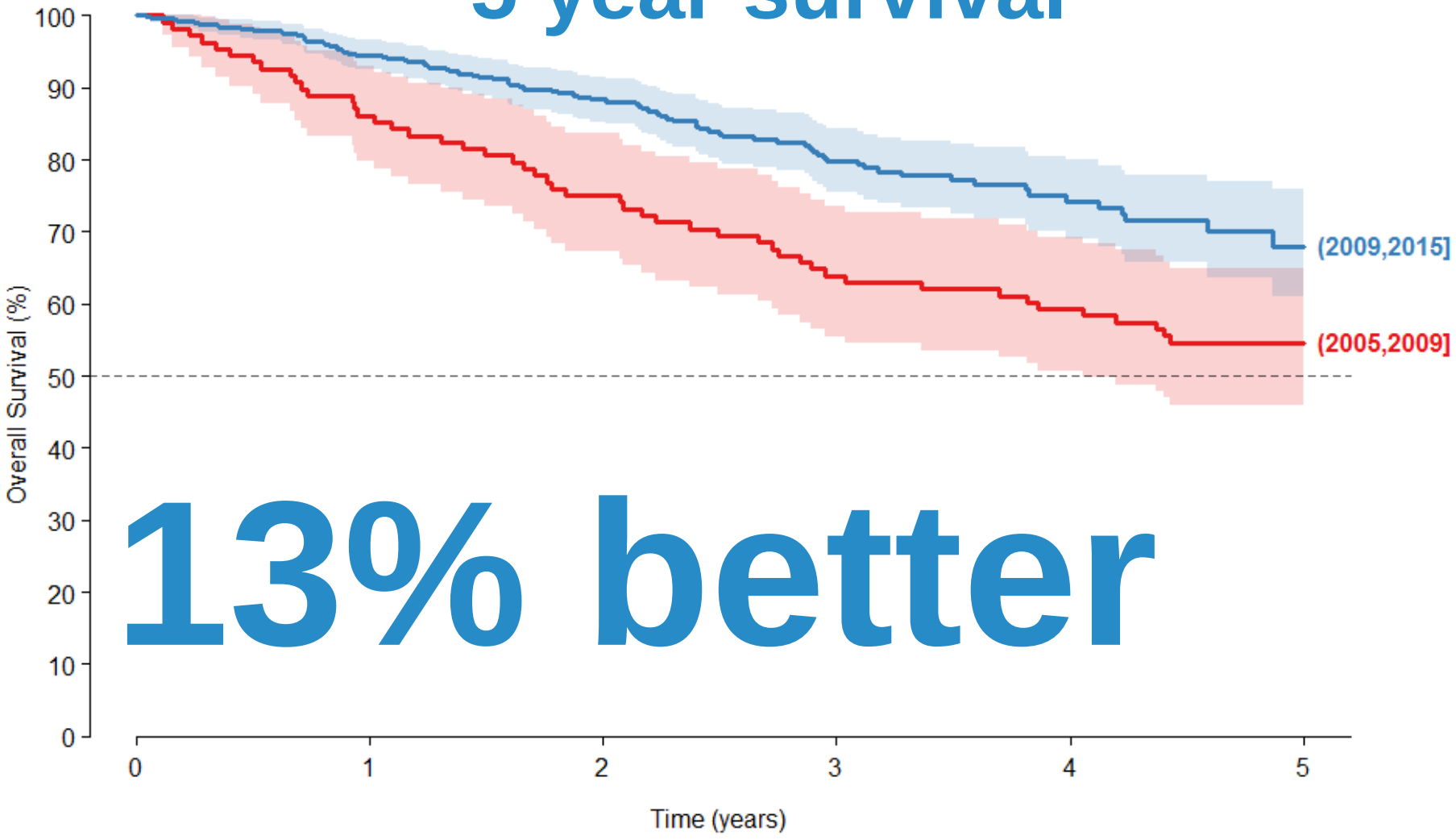
80

years to manually process

Oesophagus Cancer



Breast cancer, stage III
5 year survival



13% better

EXAMPLE 2: Predictions about individuals

Predicting survival

‘Works’

"WORKS shop assistant 3 days week"

"WORKS civil servant job centre"

"WORKS waitress"

"WORKS shop assistant"

"WORKS part time librarian"

"WORKS office manager"

‘Unfortunately’

"UNFORTUNATELY her mri scan confirmed"

"UNFORTUNATELY high risk breast carcinoma"

"UNFORTUNATELY unable cure cancer"

"UNFORTUNATELY confirmed malignant deposit "

"UNFORTUNATELY wound leaking fluid"

Predicting emergency admissions

RECOGNISE • RESUSCITATE • REFER

Unstructured text can be used to make predictions

This can be done in R

Makes huge amount of data useable

Thanks

Tom.liptrot@christie.nhs.uk