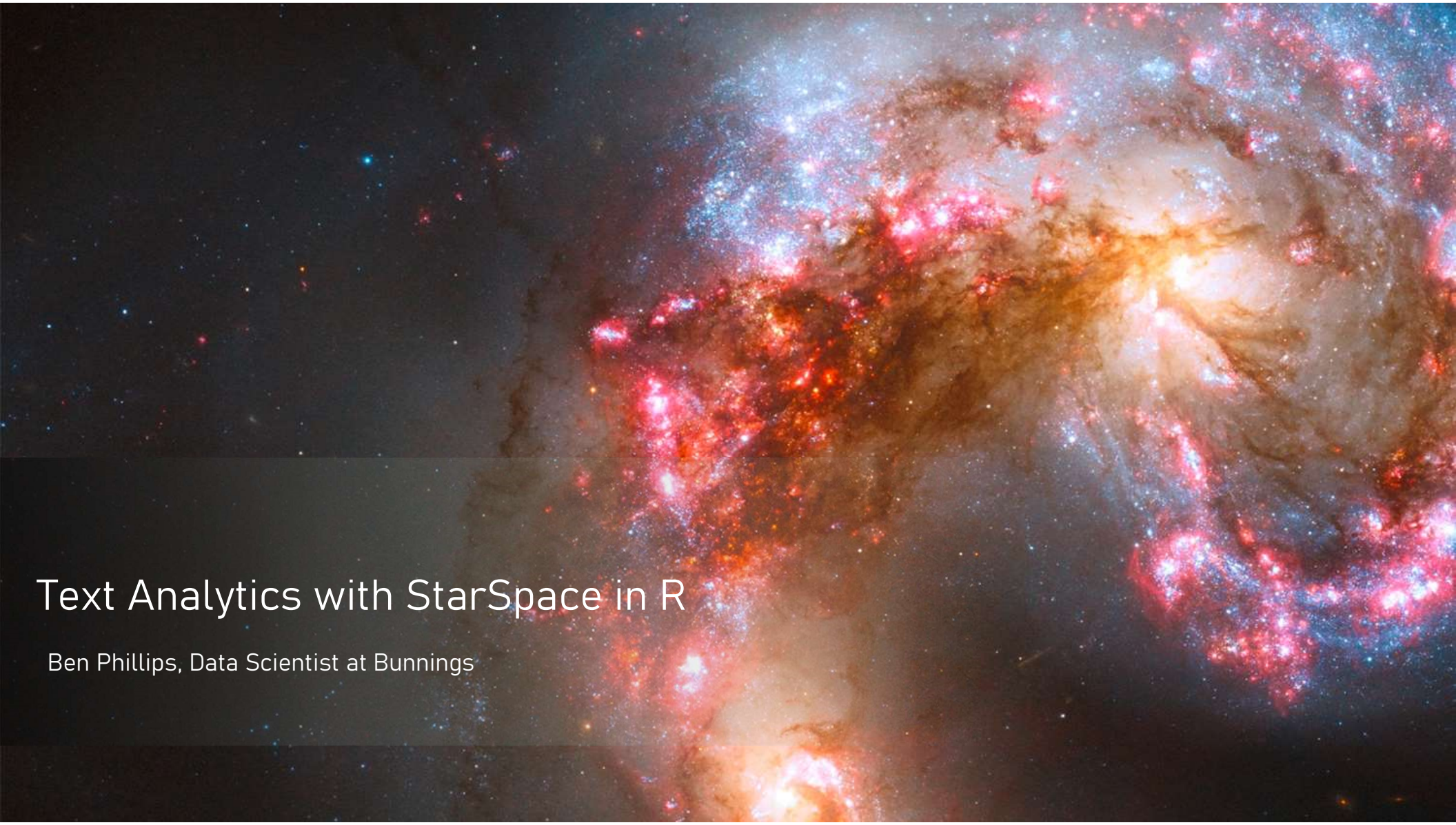


Text Analytics with StarSpace in R

Ben Phillips, Data Scientist at Bunnings

About Me

- Benjamin.phillips22@gmail.com
- Benjaminphillips22.github.io

Project Euler.net

About Project Euler

What is Project Euler?

Project Euler is a series of challenging mathematical/computer programming problems that will require more than just mathematical insights to solve. Although mathematics will help you arrive at elegant and efficient methods, the use of a computer and programming skills will be required to solve most problems.

The motivation for starting Project Euler, and its continuation, is to provide a platform for the inquiring mind to delve into unfamiliar areas and learn new concepts in a fun and recreational context.

Who are the problems aimed at?

The intended audience include students for whom the basic curriculum is not feeding their hunger to learn, adults whose background was not primarily mathematics but had an interest in things mathematical, and professionals who want to keep their problem solving and mathematics on the cutting edge.

Can anyone solve the problems?

The problems range in difficulty and for many the experience is inductive chain learning. That is, by solving one problem it will expose you to a new concept that allows you to undertake a previously inaccessible problem. So the determined participant will slowly but surely work his/her way through every problem.

Project Euler.net

Special Pythagorean triplet

Problem 9

A Pythagorean triplet is a set of three natural numbers, $a < b < c$, for which,

$$a^2 + b^2 = c^2$$

For example, $3^2 + 4^2 = 9 + 16 = 25 = 5^2$.

There exists exactly one Pythagorean triplet for which $a + b + c = 1000$.
Find the product abc.


BUNNINGS
warehouse
OPEN
7am
hire

Product Owner

Data Scientist

Safety Use Case

Data Engineer

Data Scientist/
Business Translator

Initial situation.

Data on safety incidents.

Where we want to be.

Descriptive analytics.

- Do the incidents fall into broad categories?
- What are the characteristics of those categories?

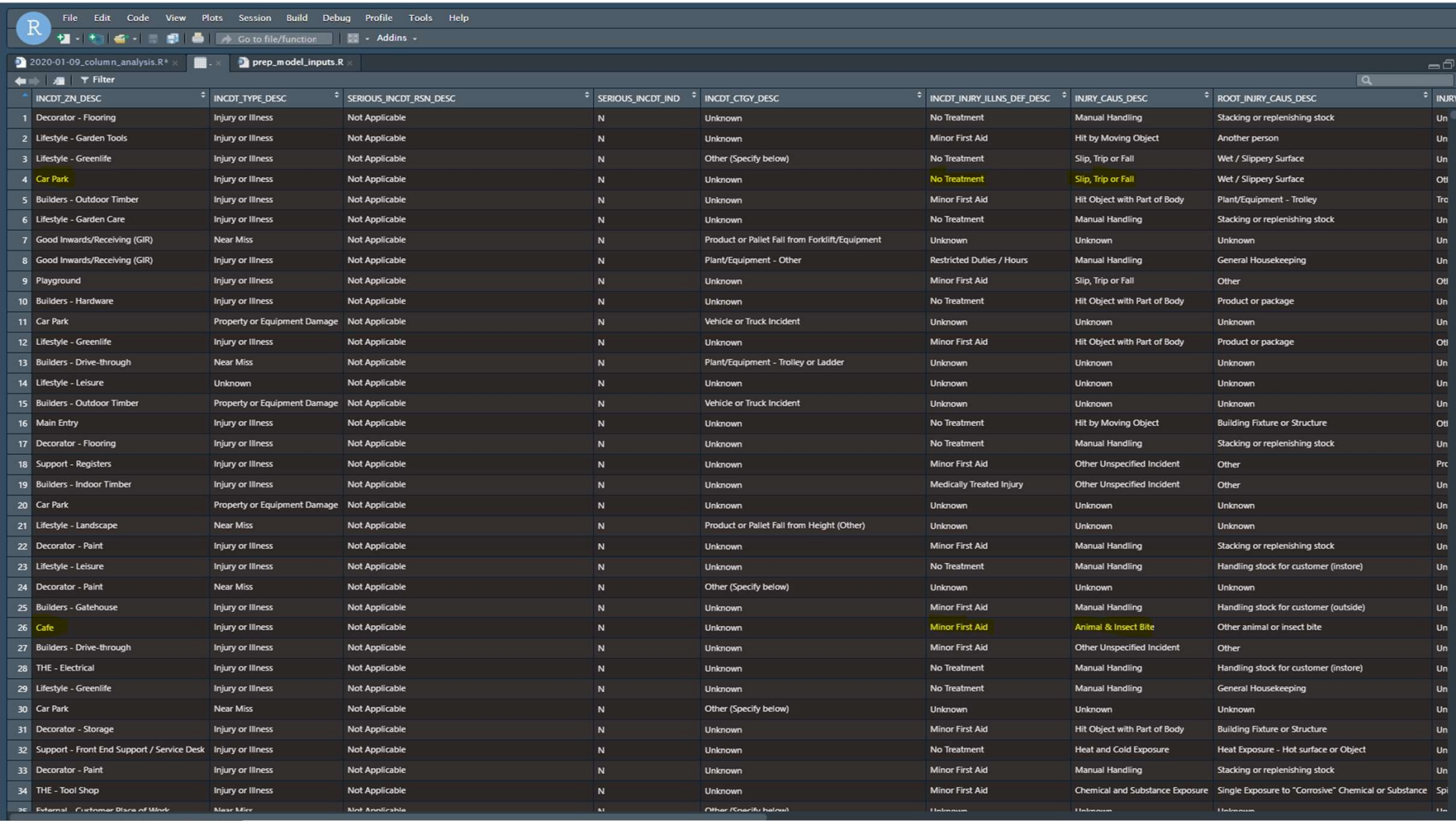
Data

"while working with another team member to put shade cloth away it was difficult to maneuver..."

"was placing items on shelf when a nearby item fell and broke on the ground..."

"tm was walking in the yard there was a big gust of wind this created a small dust storm which went into his eyes....."

"tm went to lift a gas bottle when a bee fly and..."

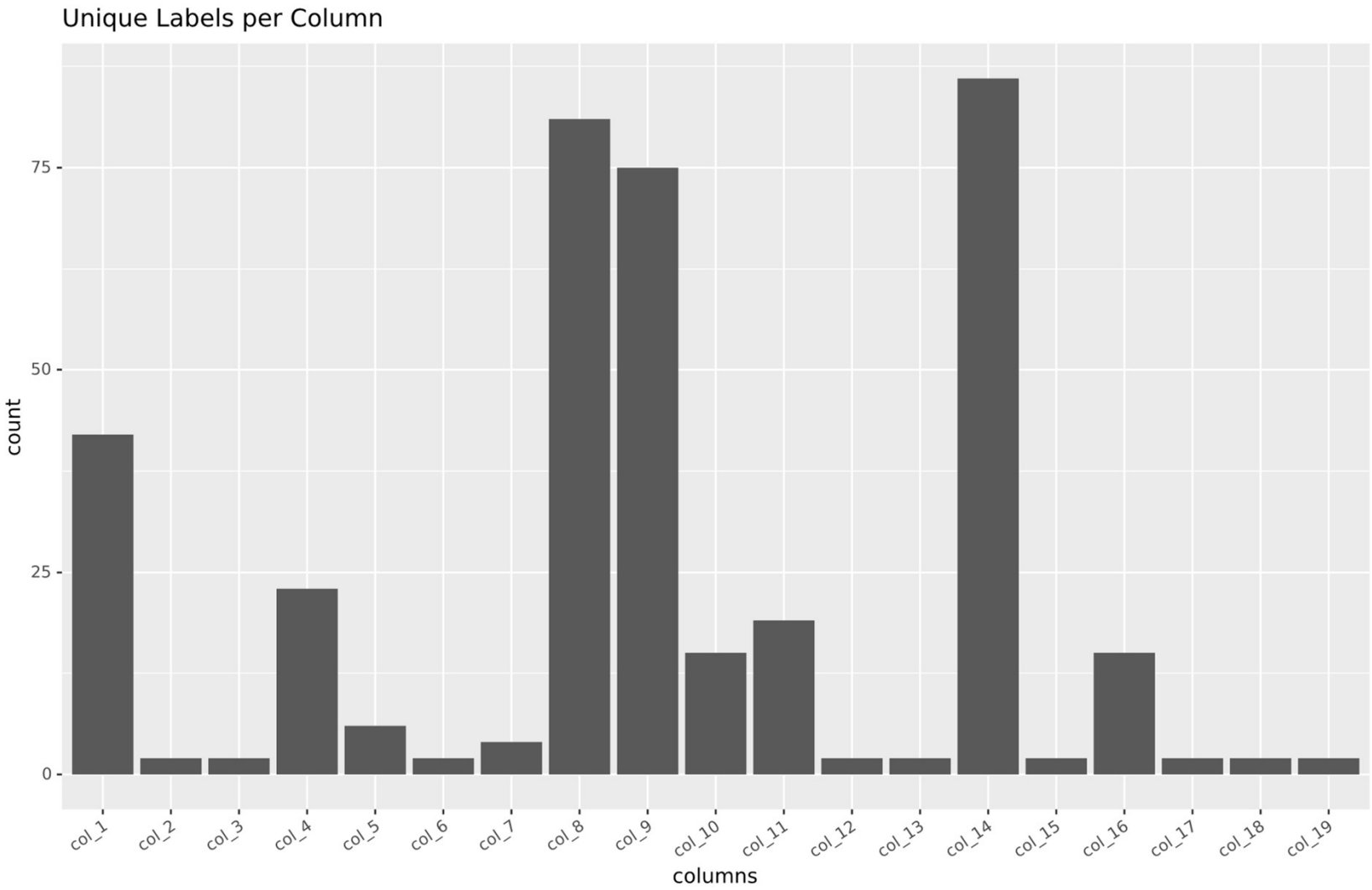

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

2020-01-09_column_analysis.R* | prep_model_inputs.R

Filter

	INCDT_ZN_DESC	INCDT_TYPE_DESC	SERIOUS_INCDT_RSN_DESC	SERIOUS_INCDT_IND	INCDT_CTGY_DESC	INCDT_INJRY_ILLNS_DEF_DESC	INJRY_CAUS_DESC	ROOT_INJRY_CAUS_DESC	INJR
1	Decorator - Flooring	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	Stacking or replenishing stock	Un
2	Lifestyle - Garden Tools	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Hit by Moving Object	Another person	Un
3	Lifestyle - Greenlife	Injury or Illness	Not Applicable	N	Other (Specify below)	No Treatment	Slip, Trip or Fall	Wet / Slippery Surface	Un
4	Car Park	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Slip, Trip or Fall	Wet / Slippery Surface	Otl
5	Builders - Outdoor Timber	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Hit Object with Part of Body	Plant/Equipment - Trolley	Tro
6	Lifestyle - Garden Care	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	Stacking or replenishing stock	Un
7	Good Inwards/Receiving (GIR)	Near Miss	Not Applicable	N	Product or Pallet Fall from Forklift/Equipment	Unknown	Unknown	Unknown	Un
8	Good Inwards/Receiving (GIR)	Injury or Illness	Not Applicable	N	Plant/Equipment - Other	Restricted Duties / Hours	Manual Handling	General Housekeeping	Un
9	Playground	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Slip, Trip or Fall	Other	Otl
10	Builders - Hardware	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Hit Object with Part of Body	Product or package	Un
11	Car Park	Property or Equipment Damage	Not Applicable	N	Vehicle or Truck Incident	Unknown	Unknown	Unknown	Un
12	Lifestyle - Greenlife	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Hit Object with Part of Body	Product or package	Otl
13	Builders - Drive-through	Near Miss	Not Applicable	N	Plant/Equipment - Trolley or Ladder	Unknown	Unknown	Unknown	Un
14	Lifestyle - Leisure	Unknown	Not Applicable	N	Unknown	Unknown	Unknown	Unknown	Un
15	Builders - Outdoor Timber	Property or Equipment Damage	Not Applicable	N	Vehicle or Truck Incident	Unknown	Unknown	Unknown	Un
16	Main Entry	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Hit by Moving Object	Building Fixture or Structure	Otl
17	Decorator - Flooring	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	Stacking or replenishing stock	Un
18	Support - Registers	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Other Unspecified Incident	Other	Pro
19	Builders - Indoor Timber	Injury or Illness	Not Applicable	N	Unknown	Medically Treated Injury	Other Unspecified Incident	Other	Un
20	Car Park	Property or Equipment Damage	Not Applicable	N	Unknown	Unknown	Unknown	Unknown	Un
21	Lifestyle - Landscape	Near Miss	Not Applicable	N	Product or Pallet Fall from Height (Other)	Unknown	Unknown	Unknown	Un
22	Decorator - Paint	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Manual Handling	Stacking or replenishing stock	Un
23	Lifestyle - Leisure	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	Handling stock for customer (instore)	Un
24	Decorator - Paint	Near Miss	Not Applicable	N	Other (Specify below)	Unknown	Unknown	Unknown	Un
25	Builders - Gatehouse	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Manual Handling	Handling stock for customer (outside)	Un
26	Cafe	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Animal & Insect Bite	Other animal or insect bite	Un
27	Builders - Drive-through	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Other Unspecified Incident	Other	Un
28	THE - Electrical	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	Handling stock for customer (instore)	Un
29	Lifestyle - Greenlife	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Manual Handling	General Housekeeping	Un
30	Car Park	Near Miss	Not Applicable	N	Other (Specify below)	Unknown	Unknown	Unknown	Un
31	Decorator - Storage	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Hit Object with Part of Body	Building Fixture or Structure	Un
32	Support - Front End Support / Service Desk	Injury or Illness	Not Applicable	N	Unknown	No Treatment	Heat and Cold Exposure	Heat Exposure - Hot surface or Object	Un
33	Decorator - Paint	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Manual Handling	Stacking or replenishing stock	Un
34	THE - Tool Shop	Injury or Illness	Not Applicable	N	Unknown	Minor First Aid	Chemical and Substance Exposure	Single Exposure to "Corrosive" Chemical or Substance	Spi
35	External - Customer Place of Work	Near Miss	Not Applicable	N	Other (Specify below)	Unknown	Unknown	Unknown	Un


Unique Labels per Column
count
75
50
25
0
col_1
col_2
col_3
col_4
col_5
col_6
col_7
col_8
col_9
col_10
col_11
col_12
col_13
col_14
col_15
col_16
col_17
col_18
col_19
columns

bunnings • Follow

bunnings Start the new year organised with our versatile garage storage solutions ✅ @pinnaclehardware's compact workbench will help declutter even the smaller working spaces. Tap to view product details. #bunnings #garagestorage

1w

1,359 likes

#TAGSPACE: Semantic Embeddings from Hashtags

Jason Weston
Facebook AI Research
jase@fb.com

Sumit Chopra
Facebook AI Research
spchopra@fb.com

Keith Adams
Facebook AI Research
kma@fb.com

Abstract

We describe a convolutional neural network that learns feature representations for short textual posts using hashtags as a supervised signal. The proposed approach is trained on up to 5.5 billion words predicting 100,000 possible hashtags. As well as strong performance on the hashtag prediction task itself, we show that its learned representation of text (ignoring the hashtag labels) is useful for other tasks as well. To that end, we present results on a document recommendation task, where it also outperforms a number of baselines.

Introduction

htags (single tokens often composed of nat- language n-grams or abbreviations, prefixed ı the character '#') are ubiquitous on social vorking services, particularly in short textual uments (a.k.a. posts). Authors use hashtags to ırse ends, many of which can be seen as labels

resented as a vector in $\mathbb{R}^n$, where n is a hyp parameter that controls capacity. The embeddi of words comprising a text are combined usin model-dependent, possibly learned function, p ducing a point in the same embedding space. similarity measure (for example, inner produ gauges the pairwise relevance of points in the e bedding space.

Unsupervised word embedding methods tr with a reconstruction objective in which the e beddings are used to predict the original text. example, word2vec tries to predict all the wo in the document, given the embeddings of s rounding words. We argue that hashtag prec tion provides a more direct form of supervisi the tags are a labeling by the author of the sali aspects of the text. Hence, predicting them n provide stronger semantic guidance than unsup vised learning alone. The abundance of hasht in real posts provides a huge labeled dataset learning potentially sophisticated models.

In this work we develop a convolutional r work for large scale ranking tasks, and apply

StarSpace: Embed All The Things!

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes and Jason Weston
Facebook AI Research

Abstract

We present StarSpace, a general-purpose neural embedding model that can solve a wide variety of problems: labeling tasks such as text classification, ranking tasks such as information retrieval/web search, collaborative filtering-based or content-based recommendation, embedding of multi-relational graphs, and learning word, sentence or document level embeddings. In each case the model works by embedding those entities comprised of discrete features and comparing them against each other – learning similarities dependent on the task. Empirical results on a number of tasks show that StarSpace is highly competitive with existing methods, whilst also being generally applicable to new cases where those methods are not.

1 Introduction

We introduce StarSpace, a neural embedding model that is general enough to solve a wide variety of problems:

- Text classification, or other labeling tasks, e.g. sentiment classification.
- Ranking of sets of entities, e.g. ranking web documents given a query.
- Collaborative filtering-based recommendation, e.g. recommending documents, music or videos.
- Content-based recommendation where content is defined with discrete features, e.g. words of documents.
- Embedding graphs, e.g. multi-relational graphs such as Freebase.
- Learning word, sentence or document embeddings.

StarSpace can be viewed as a straight-forward and efficient strong baseline for any of these tasks. In experiments it is shown to be on par with or outperforming several competing methods, whilst being generally applicable to cases where many of those methods are not

hence the "star" ("*", meaning all types) and "space" in name, and in that common space compares them aga each other. It learns to rank a set of entities, document objects given a query entity, document or object, where query is not necessarily of the same type as the items in set.

We evaluate the quality of our approach on six diffe tasks, namely text classification, link prediction in kno edge bases, document recommendation, article search, s tence matching and learning general sentence embeddi StarSpace is available as an open-source project at http //github.com/facebookresearch/Starspa

2 Related Work

Latent text representations, or *embeddings*, are vectorial r resentations of words or documents, traditionally learne an unsupervised way over large corpora. Work on ne embeddings in this domain includes (Bengio et al. 20 (Collobert et al. 2011), word2vec (Mikolov et al. 2013) more recently fastText (Bojanowski et al. 2017). In our periments we compare to word2vec and fastText as re sentative scalable models for unsupervised embeddings; also compare on the SentEval tasks (Conneau et al. 20 against a wide range of unsupervised models for sente embedding.

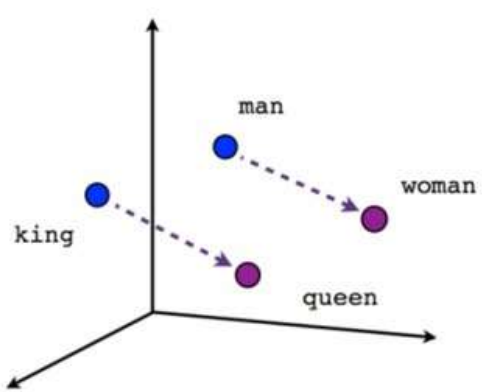
In the domain of supervised embeddings, SSI (Bai et 2009) and WSABIE (Weston, Bengio, and Usunier 20 are early approaches that showed promise in NLP and in mation retrieval tasks ((Weston et al. 2013), (Hermann e 2014)). Several more recent works including (Tang, Qin, Liu 2015), (Zhang and LeCun 2015), (Conneau et al. 20 TagSpace (Weston, Chopra, and Adams 2014) and fastT (Joulin et al. 2016) have yielded good results on classifi tion tasks such as sentiment analysis or hashtag predicti

In the domain of recommendation, embedding mo have had a large degree of success, starting from S

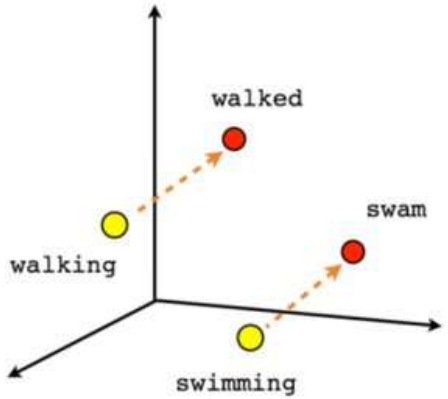
What are Embeddings?

For example

King; [0.02, -0.30, 0.66]
Queen; [0.20, -0.30, 0.55]

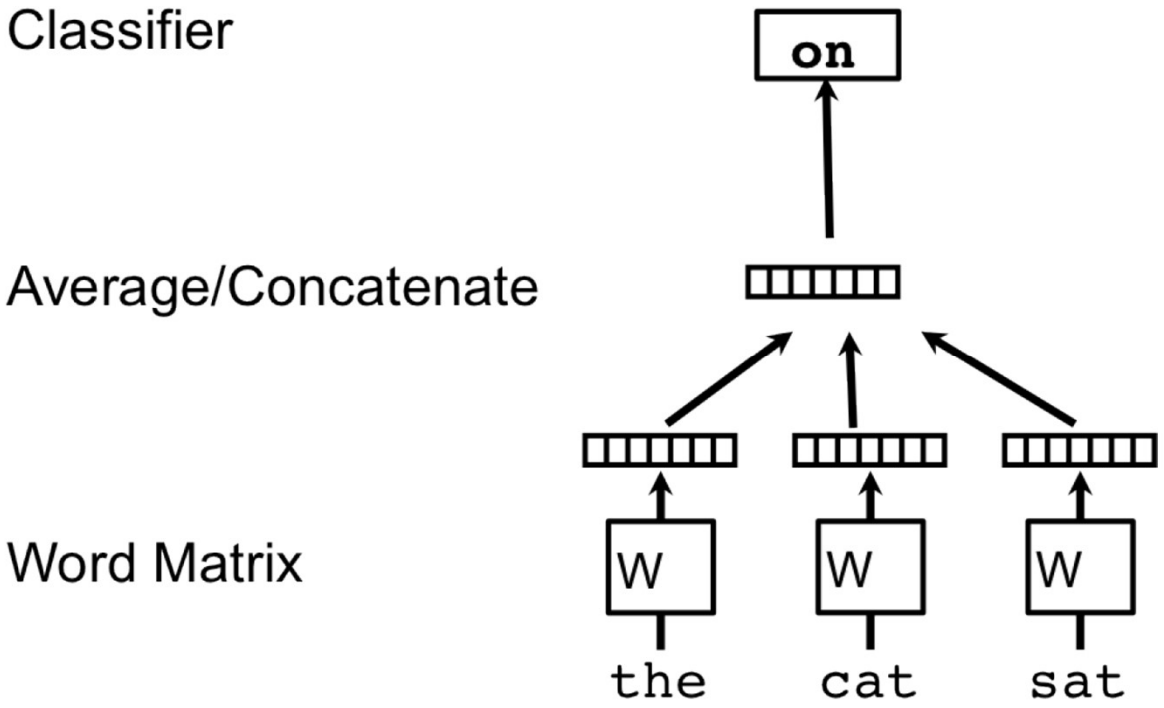


Male-Female



Verb tense

Word Embeddings



GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning

Introduction

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Getting started (Code download)

- Download the code (licensed under the Apache License, Version 2.0)
- Unpack the files: unzip GloVe-1.2.zip
- Compile the source: cd GloVe-1.2 && make
- Run the demo script: ./demo.sh
- Consult the included README for further usage details, or ask a question
- The code is also available on GitHub

Download pre-trained word vectors

- Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: http://www.opendatacommons.org/licenses/pddl/1.0/.
  - Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
  - Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
  - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
  - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip
- Ruby script for preprocessing Twitter data

Citing GloVe

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]

Baseline Sentence to Vector

the	0.04	0.34	0.78	-0.87
cat	-0.83	-0.65	0.9	0.08
sat	0.22	-0.32	-0.19	0.11
on	0.31	-0.13	0.82	0.79
	-0.065	-0.19	0.5775	0.0275

#TAGSPACE: Semantic Embeddings from Hashtags

Jason Weston
Facebook AI Research
jase@fb.com

Sumit Chopra
Facebook AI Research
spchopra@fb.com

Keith Adams
Facebook AI Research
kma@fb.com

Abstract

We describe a convolutional neural network that learns feature representations for short textual posts using hashtags as a supervised signal. The proposed approach is trained on up to 5.5 billion words predicting 100,000 possible hashtags. As well as strong performance on the hashtag prediction task itself, we show that its learned representation of text (ignoring the hashtag labels) is useful for other tasks as well. To that end, we present results on a document recommendation task, where it also outperforms a number of baselines.

1 Introduction

Hashtags (single tokens often composed of natural language n-grams or abbreviations, prefixed with the character '#') are ubiquitous on social networking services, particularly in short textual documents (a.k.a. posts). Authors use hashtags to diverse ends, many of which can be seen as labels resented as a vector in $\mathbb{R}^n$, where n is a hyperparameter that controls capacity. The embeddings of words comprising a text are combined using a model-dependent, possibly learned function, producing a point in the same embedding space. A similarity measure (for example, inner product) gauges the pairwise relevance of points in the embedding space.

Unsupervised word embedding methods train with a reconstruction objective in which the embeddings are used to predict the original text. For example, word2vec tries to predict all the words in the document, given the embeddings of surrounding words. We argue that hashtag prediction provides a more direct form of supervision: the tags are a labeling by the author of the salient aspects of the text. Hence, predicting them may provide stronger semantic guidance than unsupervised learning alone. The abundance of hashtags in real posts provides a huge labeled dataset for learning potentially sophisticated models.

In this work we develop a convolutional network for large scale ranking tasks, and apply it

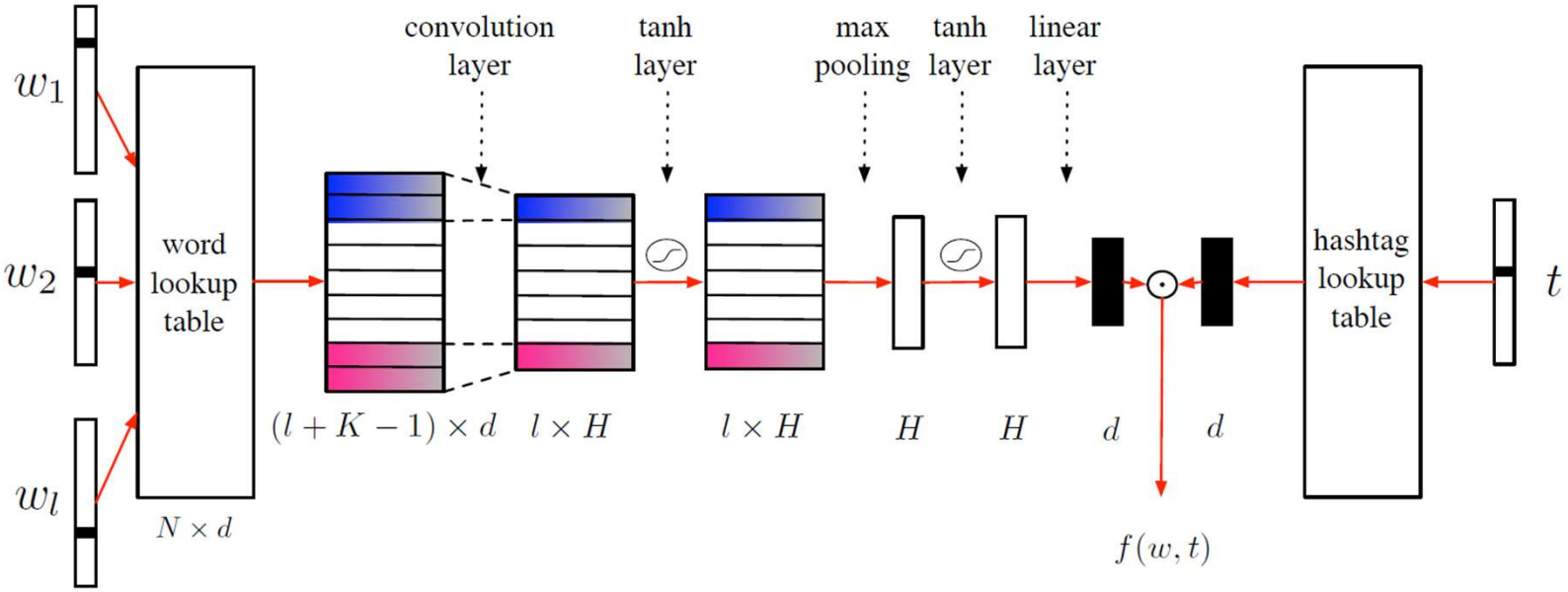
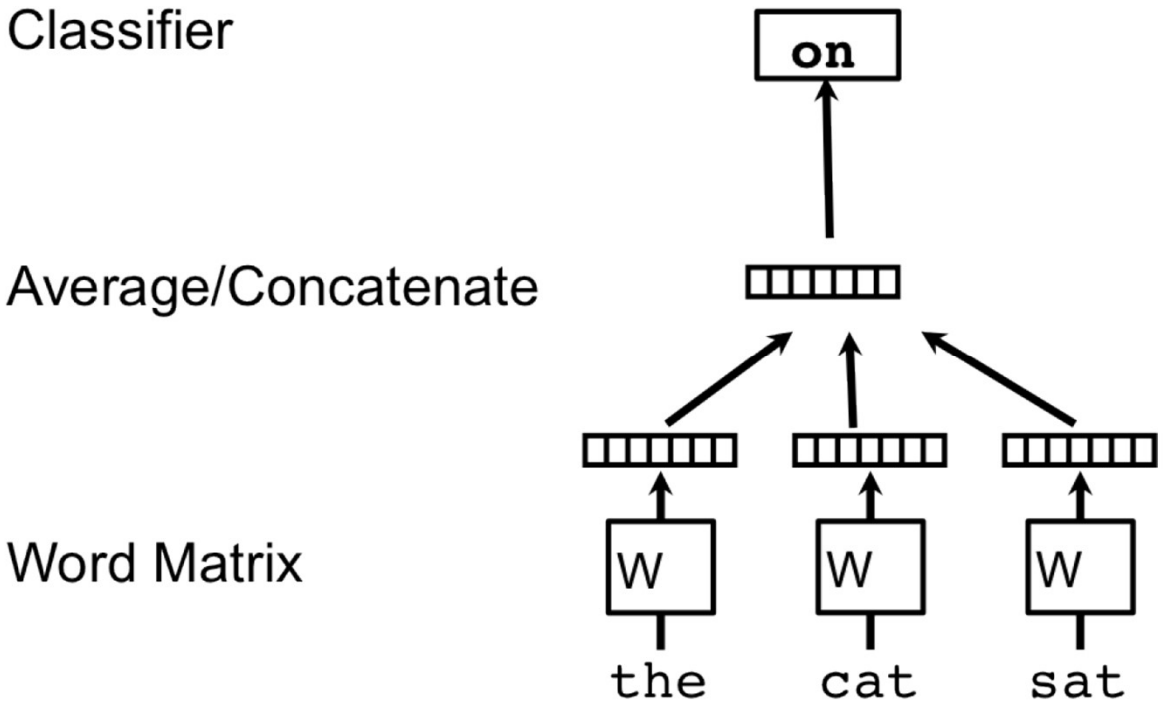
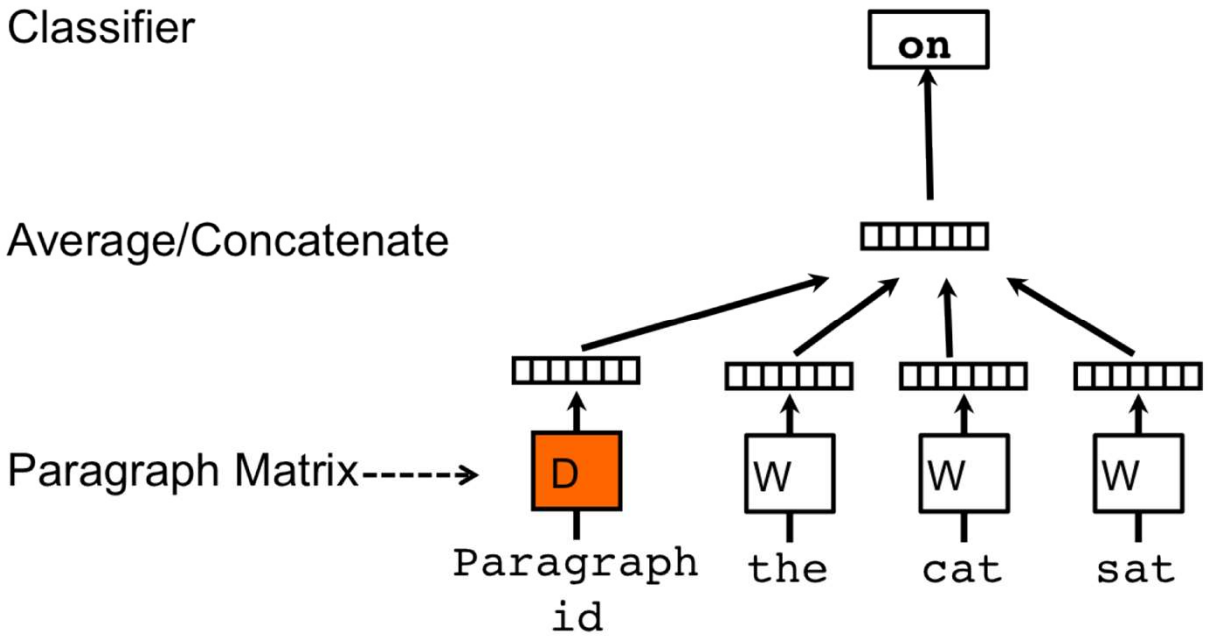


Figure 1: #TAGSPACE convolutional network $f(w, t)$ for scoring a (document, hashtag) pair.

Word Embeddings



Document Embeddings (Doc2Vec)



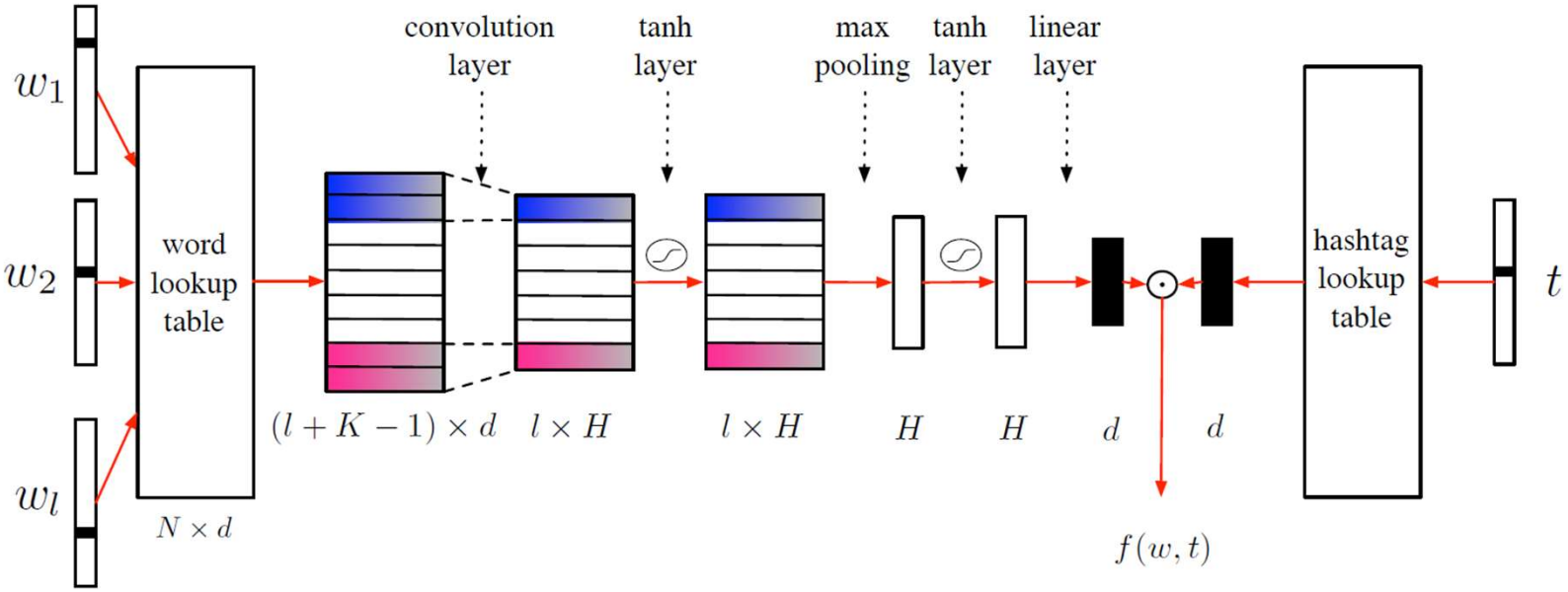


Figure 1: #TAGSPACE convolutional network $f(w, t)$ for scoring a (document, hashtag) pair.

StarSpace: Embed All The Things!

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes and Jason Weston
Facebook AI Research

Abstract

We present StarSpace, a general-purpose neural embedding model that can solve a wide variety of problems: labeling tasks such as text classification, ranking tasks such as information retrieval/web search, collaborative filtering-based or content-based recommendation, embedding of multi-relational graphs, and learning word, sentence or document level embeddings. In each case the model works by embedding those entities comprised of discrete features and comparing them against each other – learning similarities dependent on the task. Empirical results on a number of tasks show that StarSpace is highly competitive with existing methods, whilst also being generally applicable to new cases where those methods are not.

1 Introduction

We introduce StarSpace, a neural embedding model that is general enough to solve a wide variety of problems:

- Text classification, or other labeling tasks, e.g. sentiment classification.
- Ranking of sets of entities, e.g. ranking web documents given a query.
- Collaborative filtering-based recommendation, e.g. recommending documents, music or videos.
- Content-based recommendation where content is defined with discrete features, e.g. words of documents.
- Embedding graphs, e.g. multi-relational graphs such as Freebase.
- Learning word, sentence or document embeddings.

StarSpace can be viewed as a straight-forward and efficient strong baseline for any of these tasks. In experiments it is shown to be on par with or outperforming several competing methods, whilst being generally applicable to cases where many of those methods are not, hence the "star" ("*", meaning all types) and "space" in the name, and in that common space compares them against each other. It learns to rank a set of entities, documents or objects given a query entity, document or object, where the query is not necessarily of the same type as the items in the set.

We evaluate the quality of our approach on six different tasks, namely text classification, link prediction in knowledge bases, document recommendation, article search, sentence matching and learning general sentence embeddings. StarSpace is available as an open-source project at `https://github.com/facebookresearch/Starspace`.

2 Related Work

Latent text representations, or *embeddings*, are vectorial representations of words or documents, traditionally learned in an unsupervised way over large corpora. Work on neural embeddings in this domain includes (Bengio et al. 2003), (Collobert et al. 2011), word2vec (Mikolov et al. 2013) and more recently fastText (Bojanowski et al. 2017). In our experiments we compare to word2vec and fastText as representative scalable models for unsupervised embeddings; we also compare on the SentEval tasks (Conneau et al. 2017) against a wide range of unsupervised models for sentence embedding.

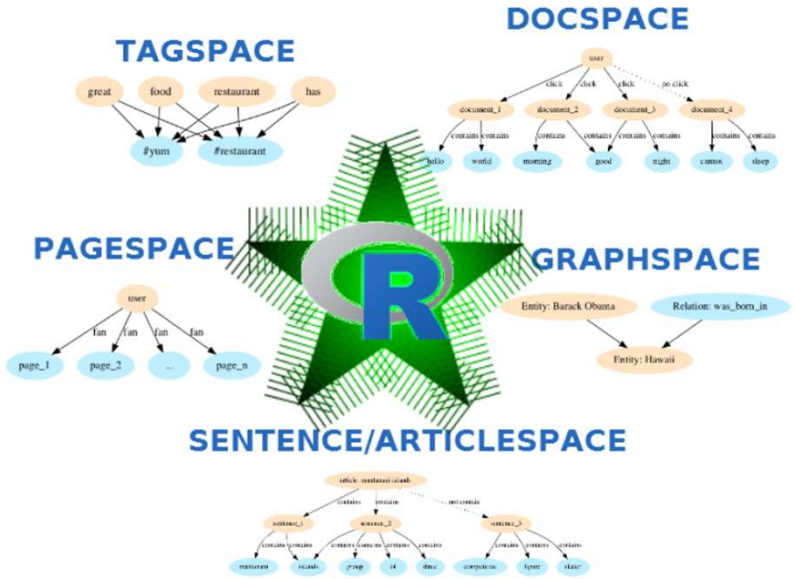
In the domain of supervised embeddings, SSI (Bai et al. 2009) and WSABIE (Weston, Bengio, and Usunier 2011) are early approaches that showed promise in NLP and information retrieval tasks ((Weston et al. 2013), (Hermann et al. 2014)). Several more recent works including (Tang, Qin, and Liu 2015), (Zhang and LeCun 2015), (Conneau et al. 2016), TagSpace (Weston, Chopra, and Adams 2014) and fastText (Joulin et al. 2016) have yielded good results on classification tasks such as sentiment analysis or hashtag prediction.

In the domain of recommendation, embedding models have had a large degree of success, starting from SVD

ruimtehol: R package to Embed All the Things! using StarSpace

This repository contains an R package which wraps the StarSpace C++ library (https://github.com/facebookresearch/StarSpace), allowing the following:

- Text classification
- Learning word, sentence or document level embeddings
- Finding sentence or document similarity
- Ranking web documents
- Content-based recommendation (e.g. recommend text/music based on the content)
- Collaborative filtering based recommendation (e.g. recommend text/music based on interest)
- Identification of entity relationships



Getting the Data Ready

```
#' Clean text for modelling
#'
#' @export
#' @description cleans text by removing apostrophe's in words, removes all
#'     punctuations and line feeds, digits and any extraneous spaces
#' @param x a character vector
#' @return cleaned character vector
cleanText <- function(x){
    output <- x %>%
        ## remove "'" from words to change "didn't" to "didnt"
        stringr::str_replace_all(., "'", "") %>%
        ## remove punctuation, double quotes and line feeds
        stringr::str_replace_all(., "[\\p{P}\\p{S}\r\n\t]", " ") %>%
        ## remove strings of numbers
        stringr::str_replace_all(., "[0-9]+", " ") %>%
        ## seperate by only one space
        stringr::str_replace_all(., "[ ]{2,}", " ") %>%
        ## remove any whitespace at beginning or end of line
        stringr::str_trim(.) %>%
        ## lower case everything
        tolower(.)

    return(output)
}
```

Getting the Data Ready

```
288  287,forkilt,forklift
289  288,forklifted,forklifted
290  289,forkklift,forklift
291  290,forklfit,forklift
292  291,forklft,forklift
293  292,forklifter,forklift
294  293,forklfting,forklifting
295  294,forklif,forklift
296  295,forklife,forklift
297  296,forkliftiting,forklifting
298  297,forkliftt,forklift
299  298,forkligt,forklift
300  299,forkligting,forklifting
301  300,forkliift,forklift
302  301,forklilft,forklift
303  302,forklioft,forklift
304  303,forklirt,forklift
305  304,forklist,forklift
306  305,forklit,forklift
307  306,forkllift,forklift
308  307,forkllioft,forklift
309  308,forksjust,forks just
310  309,forksthey,forks they
311  310,forksw,forks
312  311,forktine,fork tine
```

Running the Algorithm

```
# set seed for tagspace
set.seed(321)
model <- ruimtehol::embed_tagspace(x = trainList$x,
                                   y = trainList$y,
                                   dim = 30,
                                   early_stopping = 0.8,
                                   validationPatience = 10,
                                   epoch = 100,
                                   lr = 0.01,
                                   loss = "hinge",
                                   negSearchLimit = 10,
                                   adagrad = FALSE,
                                   initRandSd = 0.01,
                                   ngrams = 1,
                                   minCount = 20,
                                   thread = 6)
```

Evaluating TagSpace

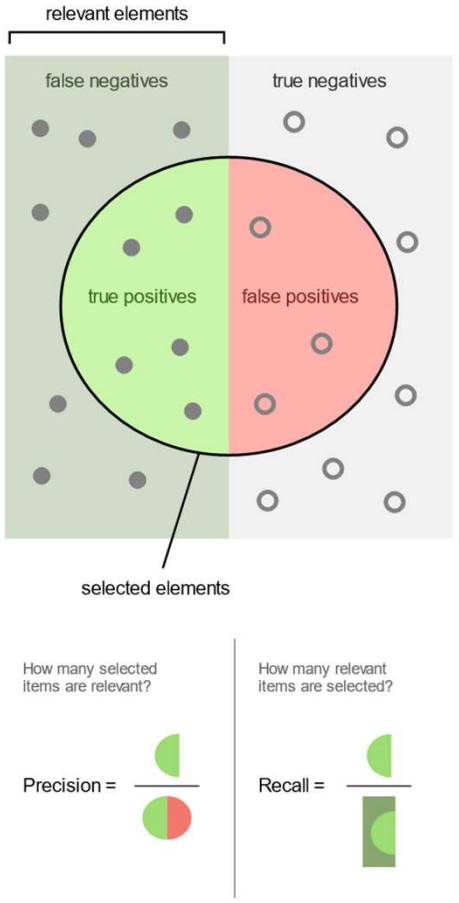
Precision at 1

Recall at 10

Mean Rank of True Labels

1 #Bunnings
2 #summer
3 #garagestorage
4 #shed

bunnings Start the new year organised with our versatile garage storage solutions ✅ @pinnaclehardware's compact workbench will help declutter even the smaller working spaces. Tap to view product details. #bunnings #garagestorage



Where we are.

TagSpace model to convert incident descriptions to embeddings

Where we want to be.

Descriptive analytics.

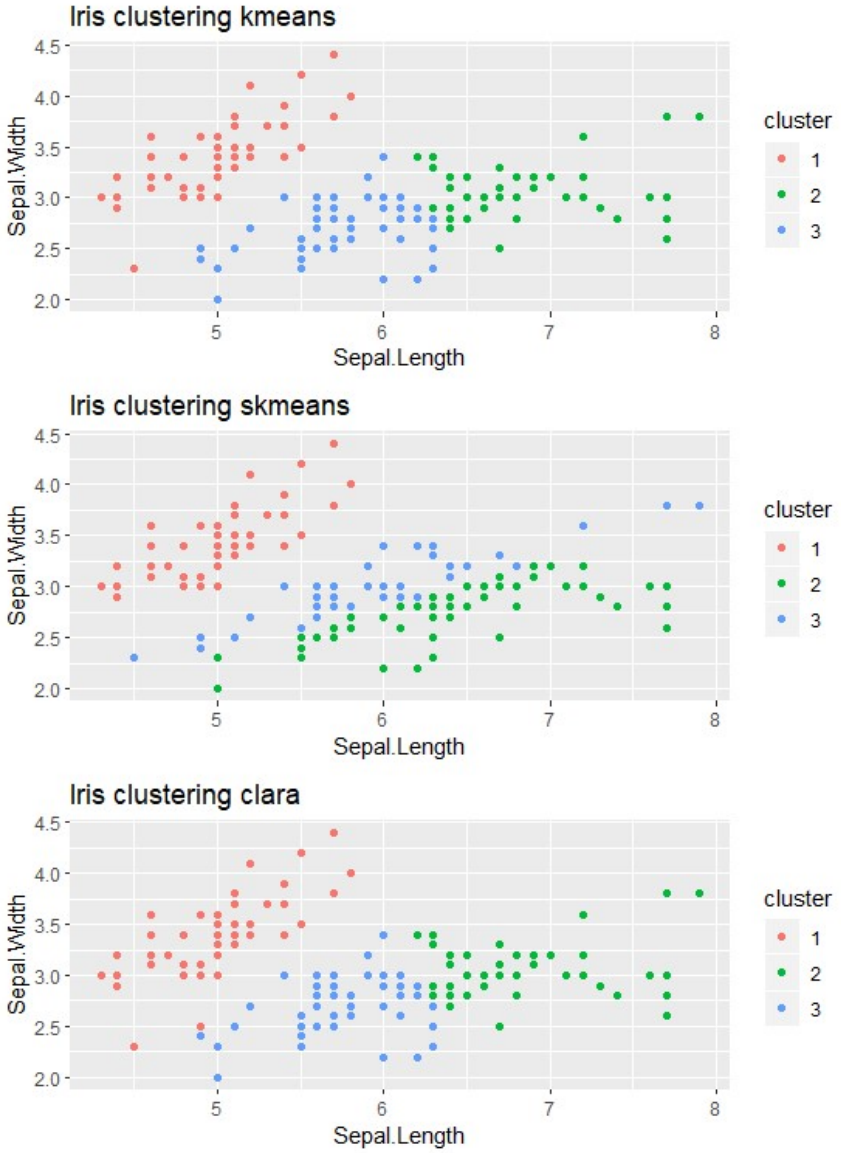
- Do the incidents fall into broad categories?
- What are the characteristics of those categories?

What is Clustering?

kmeans – partition the points into *k* groups such that the sum of squares from points to the assigned cluster centres is minimized

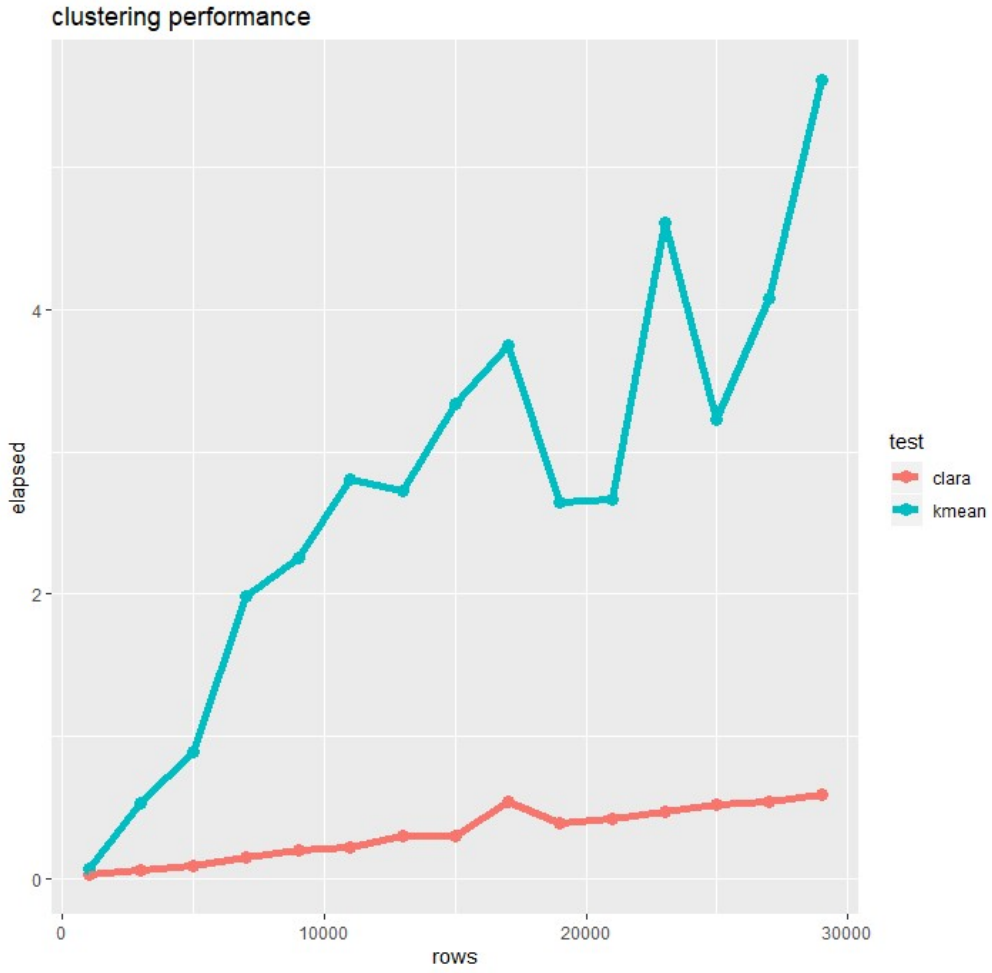
skmeans (spherical kmeans) – similar to kmeans but uses cosine similarity distance instead.

clara (Clustering Large Applications) – similar to kmeans but takes samples of the data to be more resource efficient.



What is Clustering?

kmeans vs clara –
clara is significantly less computational expensive compared to kmeans.



Clustering

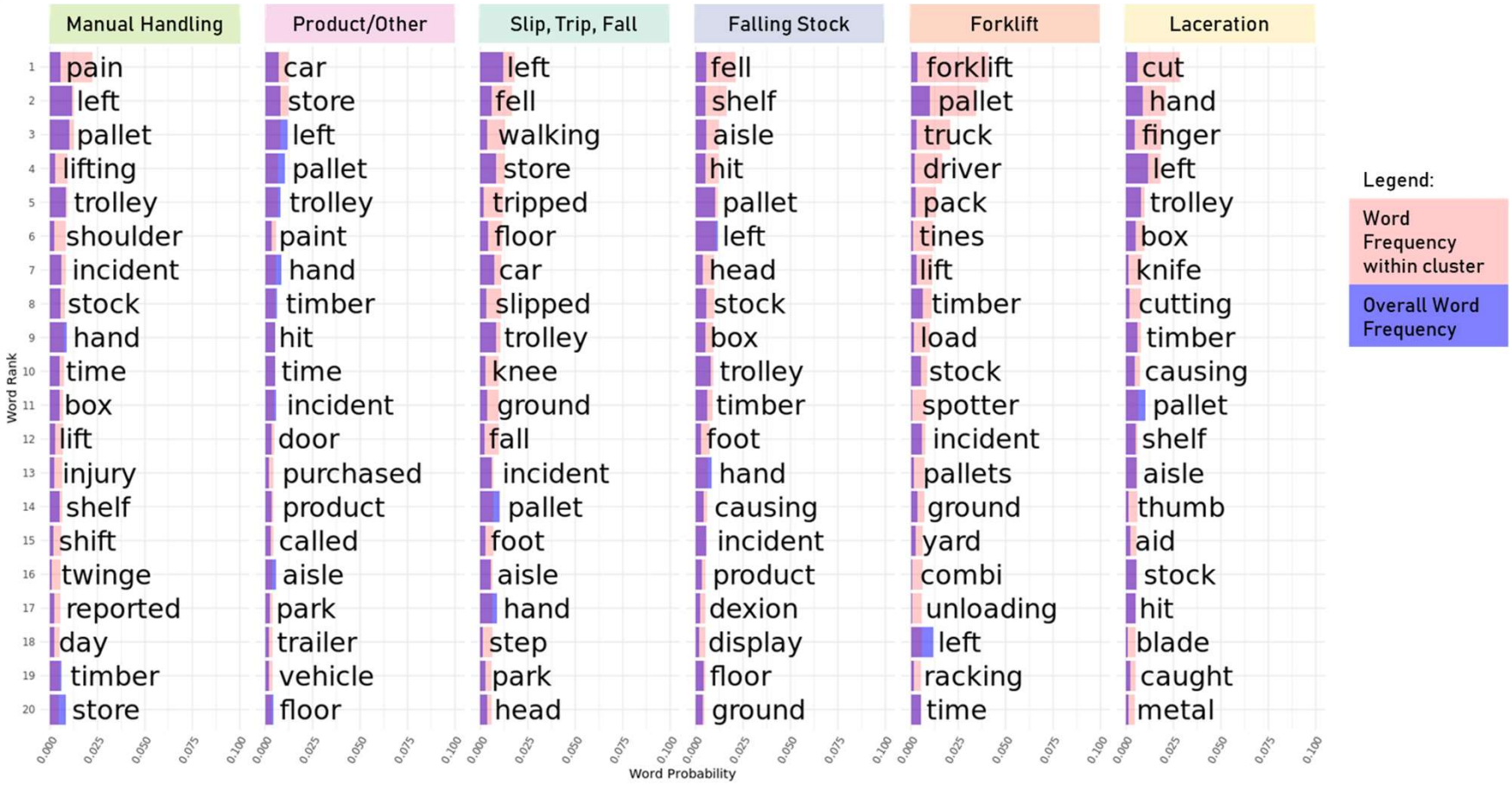
```
## get the functions for clustering options
cluster_list <- list(

    kmeans = stats::kmeans(
        x = dt,
        centers = 7,
        iter.max = 50),

    clara = cluster::clara(
        x = dt,
        k = 7,
        samples = 100),

    skmeans = skmeans::skmeans(
        x = dt,
        k = 7)
)
```

Cluster Evaluations (Quantitative)

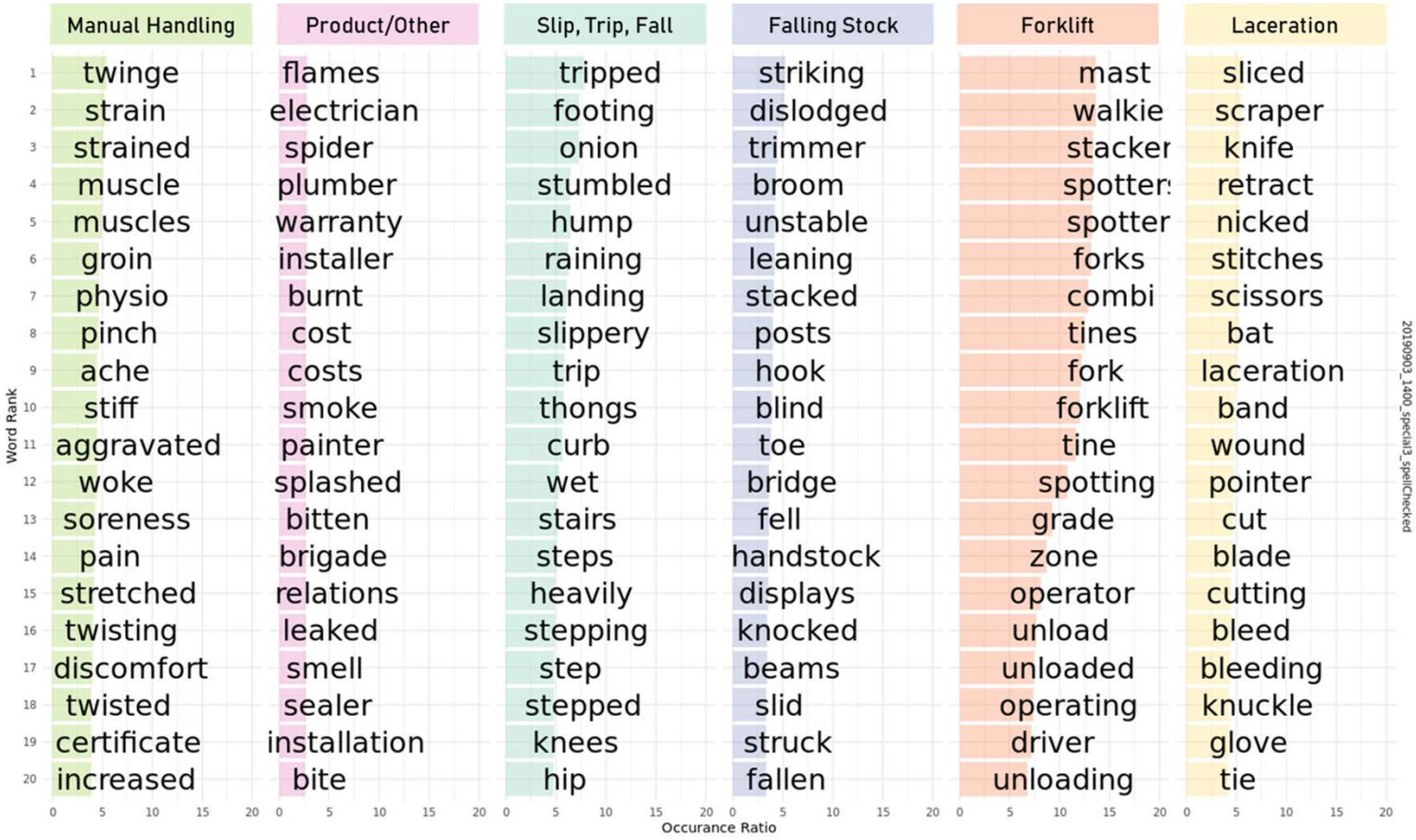
Silhouette – a measurement of how close a point is to every other point in the same cluster compared to every point in the next closest cluster.

Dispersion – a measurement of how compact the points in a cluster are.

Cluster Evaluations (Qualitative)



Cluster Evaluations (Qualitative)



Where are we.

Descriptive analytics.

What next?

Predictive and Prescriptive Analytics

- Can we predict safety incidents?
- Can we prevent incidents from occurring or minimise their impact?


WATCH OUT FOR
FORKLIFTS