

Text-Denoising

mit Go(lang)

Bedcon 2015 – Dennis Kluge – Charité Berlin

<your-logo-goes-here>

FORSCHUNG DURCH VERNETZUNG

Das Forschungsprojekt DataFlex hat zum Ziel, anonyme medizinische Daten miteinander zu vernetzen, auszuwerten und adäquat zu vermitteln. Dank **innovativer technischer Verfahren** ist der Datenschutz beim Zugriff auf die enorme Datenmenge garantiert.

Kontakt aufnehmen



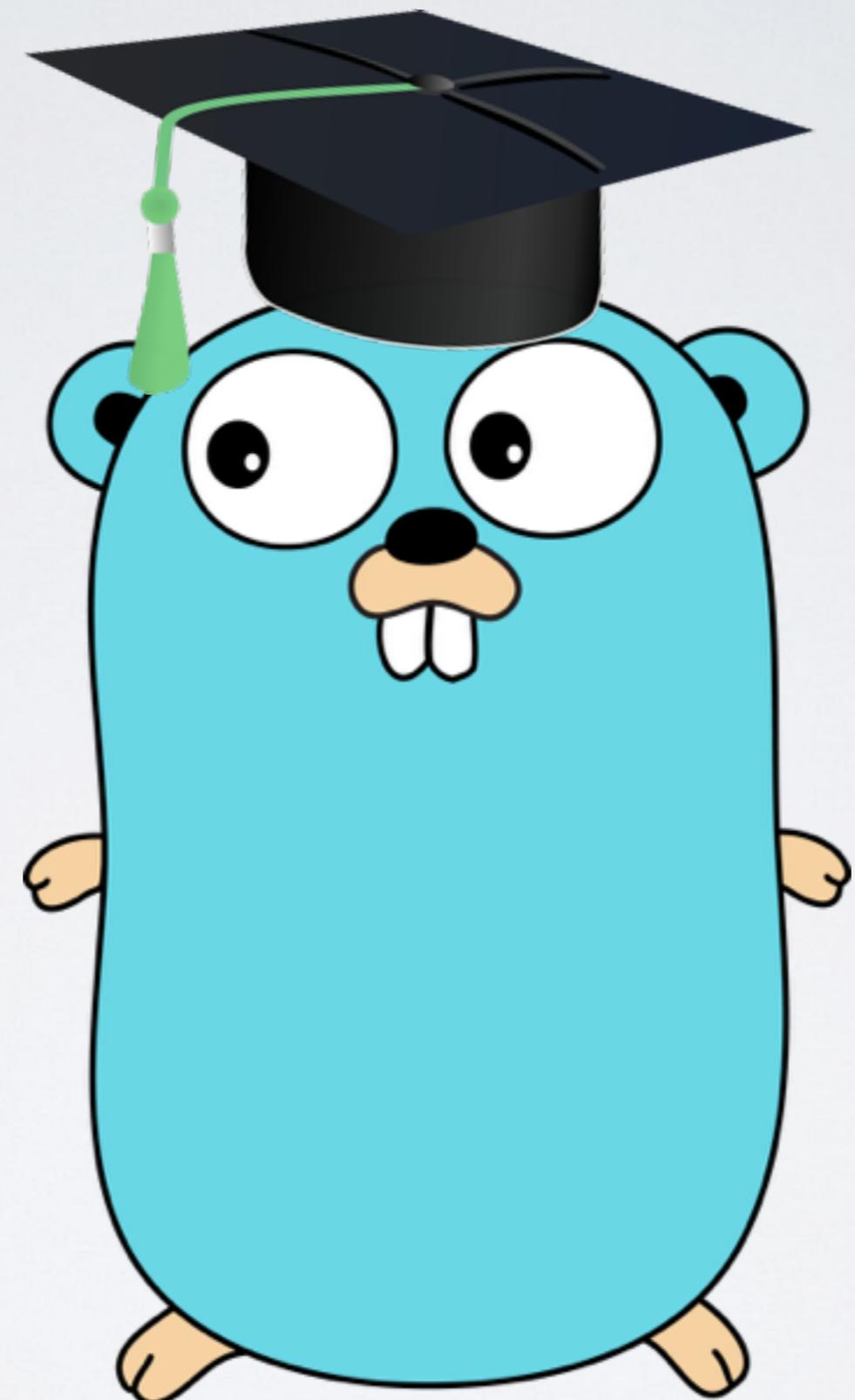
dataflex-science.de

THEMEN



Validierung eines neuartigen Datenmanagements

Innerhalb des Forschungsprojekts OpEN.SC der Forschungsgruppe Digitale Pathologie wurde in der Charité-Universitätsmedizin Berlin ein neuartiges Datenmanagement entwickelt. Es ermöglicht, ein Netzwerk von Daten aufzubauen ohne bestimmte Profilinformationen oder personenbezogene Daten zu speichern. Das





Berlin Expert Days 2015 | Dennis Kluge | Folie 4

NLP

Natural Language Processing



Die **@bedcon** ist die großartigste
#Konferenz des Jahres. 😍

<http://bedcon.org>

Die bedcon ist die großartigste
Konferenz des Jahres

die bedcon ist die großartigste
konferenz des jahres

[“di”, “ie”, “e_”, “_b”, “be”, “ed”...]



1

2

3

4

5

GO(LANG)

- 2009 - erschienen ... 2012 - Version 1.0
- kompiliert, stark typisiert, imperativ, strukturiert
- optimiert für Nebenläufigkeit
- Garbage Collection
- C angelehnte Syntax

STRINGS

- Go source code is always UTF-8.
- A string holds arbitrary bytes.
- A string literal, absent byte-level escapes, always holds valid UTF-8 sequences.
- Those sequences represent Unicode code points, called runes.
- No guarantee is made in Go that characters in strings are normalized.
- <https://blog.golang.org/strings>

```
1 var awesomeTweet string = "Die @bedcon ist die großartigste  
2 #Konferenz des Jahres. 😍  
3 http://bedcon.org"
```

```
1 var awesomeTweet string = "Die @bedcon ist die großartigste  
2 #Konferenz des Jahres. 😍  
3 http://bedcon.org"
```



**Boolean, Numeric,
String, Array, Slice,
Map, Interface, Map, Channel**

```
1 awesomeTweet := "Die @bedcon ist die großartigste  
2 | #Konferenz des Jahres. 😍  
3 | http://bedcon.org"
```

```
1 awesomeTweet := "Die @bedcon ist die großartigste  
2 | #Konferenz des Jahres. 😍  
3 | http://bedcon.org"
```



Type Inference

```
1 type tweet struct {
2     text string
3     metadata map[string]string
4 }
5
6 tweet := &tweet{text: "foobar",
7                     metadata: map[string]string{"location": "berlin"}}
```

```
1 lowerCasedTweet := strings.ToLower(awesomeTweet)
```

```
1 lowerCasedTweet := strings.ToLower(awesomeTweet)
```



Package

```
1 package strings  
2  
3 func ToLower(text string) {...}
```

```
1 package strings  
2  
3 func ToLower(text string) {...}
```



Großbuchstabe deklariert public

```
1 fmt.Printf("Lower Case: %s \n", lowerCasedTweet)
```

```
1 fmt.Printf("Lower Case: %s \n", lowerCasedTweet)
```



C-Style

```
1 tweetWithoutHashtags := strings.Replace(lowerCasedTweet, "#", "", -1)
```

```
1 urlRegexp := regexp.MustCompile(  
2     `((( [A-Za-z]{3,9}:(?:\/\/)?)(?:[-;:&=\$\w]+@[  
3         [A-Za-z0-9.-]+|(?::www.|[-;:&=\$\w]+@[  
4         [A-Za-z0-9.-]+)((?:\/[\+\~%\./.\w_-]*)  
5         ?\??(?:[-\+=&%@\.\w_]*))#?(?:[\w]*)?)`)  
6 withoutURL := urlRegexp.ReplaceAllString(withoutMention, "")
```

MATCHING URLs

- mathiasbynens.be/demo/url-regex
- stackoverflow.com/questions/161738/what-is-the-best-regular-expression-to-check-if-a-string-is-a-valid-url

```
1 func deleteEmoticons(text string) string {
2     normalizedString := bytes.NewBufferString("")
3     for _, runeValue := range text {
4         if runeValue >= rune(0x1F600) && runeValue <= rune(0x1F64F) {
5             normalizedString.WriteString("")
6         } else {
7             normalizedString.WriteRune(runeValue)
8         }
9     }
10    return normalizedString.String()
11 }
```

```
1 func extractNGrams(text string, steps int) []string {  
2     var ngrams []string  
3     for i := 0; i < (len(text) - steps + 1); i++ {  
4         ngrams = append(ngrams, text[i:i+steps])  
5     }  
6     return ngrams  
7 }  
8
```

```

1 package main
2
3 import (
4     "bytes"
5     "fmt"
6     "regexp"
7     "strings"
8 )
9
10 func deleteEmoticons(text string) string {
11     normalizedString := bytes.NewBufferString("")
12     for _, runeValue := range text {
13         if runeValue >= rune(0x1F600) && runeValue <= rune(0x1F64F) {
14             normalizedString.WriteString("")
15         } else {
16             normalizedString.WriteRune(runeValue)
17         }
18     }
19     return normalizedString.String()
20 }
21
22 func extractNGrams(text string, steps int) []string {
23     var ngrams []string
24     for i := 0; i < (len(text) - steps + 1); i++ {
25         ngrams = append(ngrams, text[i:i+steps])
26     }
27     return ngrams
28 }
29
30 func main() {
31     awesomeTweet := "Die @bedcon ist die großartigste #Konferenz des Jahres. 😊 http://bedcon.org"
32     lowerCasedTweet := strings.ToLower(awesomeTweet)
33     fmt.Printf("Lower Case: %v \n", lowerCasedTweet)
34
35     withoutHashtag := strings.Replace(lowerCasedTweet, "#", "", -1)
36     fmt.Printf("Without Hashtag: %v \n", withoutHashtag)
37
38     withoutMention := strings.Replace(withoutHashtag, "@", "", -1)
39     fmt.Printf("Without Mention: %v \n", withoutMention)
40
41     urlRegexp := regexp.MustCompile(`((([A-Za-z]{3,9}:(?:\/\/)?(?:[-;:&=\$\w]+@)?([A-Za-z0-9.-]+|(?:www.|[-;:&=\$\w]+@)[A-Za-z0-9.-]+)((?:\/\/|\+\~\//|\w_-)*|(\?|\?(\[-\+\&@\.\w_]*|\#?\?(\[\w]*))?)`))
42     withoutURL := urlRegexp.ReplaceAllString(withoutMention, "")
43     fmt.Printf("Without URL: %v \n", withoutURL)
44
45     withoutEmoticons := deleteEmoticons(withoutURL)
46     fmt.Printf("Without Emoticons: %v \n", withoutEmoticons)
47
48     nGrams := extractNGrams(withoutEmoticons, 2)
49     fmt.Printf("NGrams: %v \n", nGrams)
50 }

```

REFACTORING IS KEY

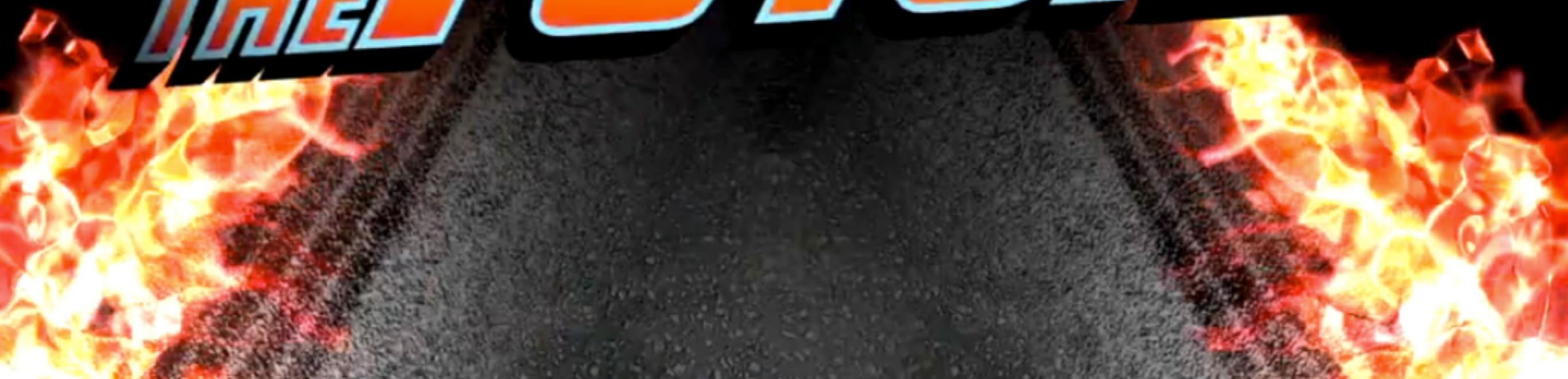


```
1 chain := NewChainNormalizer(  
2     NewLowerCaseNormalizer(),  
3     NewUnicodeRangeNormalizerFromChart(normalizers.EmoticonChart, ""),  
4     NewUrlReplacementNormalizer(""),  
5     NewRangeTableNormalizer(unicode.Latin, " "),  
6     NewStringReplacementNormalizer([]string{"?", ".", ",", ","}, ""),  
7     NewWhitespaceNormalizer())
```



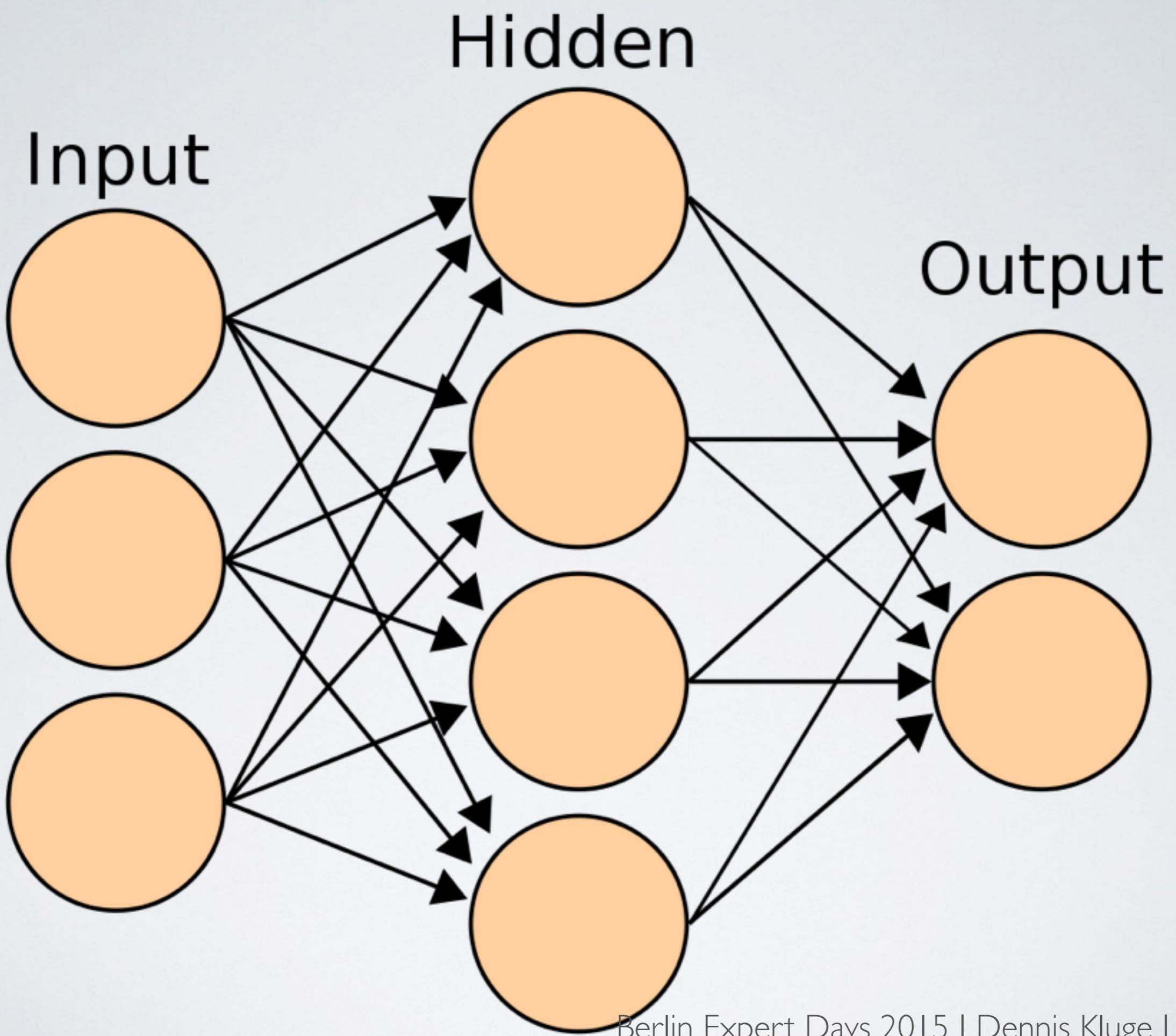
github.com/horstmumpitz/bedcon2015

BACK TO THE FUTURE™



BAG OF BIGRAMS

Bigram	Tweet 1	Tweet 2	Tweet 3
di	1	2	0
ie	0	5	4
e_	3	0	9
...			





–dennis.kluge@charite.de
– @HorstMumpitz