Northeastern University

INFO 6105
Data Sci Eng Mth & Tools
Lecture 3 udpipe addendum for non-English text

16 January 2018

Universal dependencies integrated Treebank

- Universal Dependencies (UD) is a framework for cross-linguistically consistent grammatical annotation and an open community effort with over 200 contributors producing more than 100 treebanks in over 60 languages
  - http://universaldependencies.org/
- **Udpipe** provides language-agnostic ‘tokenization’ and ‘parts of speech tagging’, of raw text in many languages, including Chinese and Hindi
- We speak a lot of Chinese and Hindi in class, so this is perfect

小娃撑小艇

偷采白蓮回

不解藏蹤跡

浮萍一道開

मैं तन्हा हूँ मुझे तन्हा ही रहने दो

देखकर मेरे बहते आंसू

तुम अपने लहू न बहने दो

मैं आपका दीवाना हूँ

मुझे बस अपना पागल रहने दो

3

Universal Dependencies

- Project that seeks to develop cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective
- The annotation scheme is based on (universal) Stanford dependencies (de Marneffe et al., 2006, 2008, 2014), Google universal part-of-speech (POS) tags (Petrov et al., 2012), and the Interset interlingua for morphosyntactic tagsets (Zeman, 2008)
- Pre-trained Universal Dependencies 2.0 models on all UD treebanks are made available for more than 50 languages:
  - afrikaans, ancient_greek-proiel, ancient_greek, arabic, basque, belarusian, bulgarian, catalan, chinese, coptic, croatian, czech-cac, czech-cltt, czech, danish, dutch-lassysmall, dutch, english-lines, english-partut, english, estonian, finnish-ftb, finnish, french-partut, french-sequoia, french, galician-treegal, galician, german, gothic, greek, hebrew, hindi, hungarian, indonesian, irish, italian, japanese, kazakh, korean, latin-ittb, latin-proiel, latin, latvian, lithuanian, norwegian-bokmaal, norwegian-nynorsk, old_church_slavonic, persian, polish, portuguese-br, portuguese, romanian, russian-syntagrus, russian, sanskrit, serbian, slovak, slovenian-sst, slovenian, spanish-ancora, spanish, swedish-lines, swedish, tamil, turkish, ukrainian, urdu, uyghur, vietnamese
- https://universaldependencies.org/introduction.html
- https://ufal.mff.cuni.cz/udpipe/users-manual

Neural network

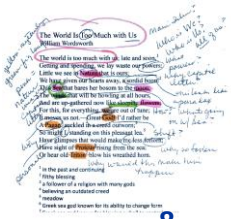
- http://ufal.mff.cuni.cz/parsito
- https://github.com/ufal/parsito/blob/master/src/network/neural_network.cpp

Part 1

LANGUAGE ENCODINGS

Annotating with pretrained language models

- **`Udpipe`** provides language-agnostic ‘tokenization’ and ‘parts of speech tagging’, of raw text in many languages, including Chinese and Hindi.
- `library(udpipe)`
- `model <- udpipe_download_model(language = "english")`
- `# When you download the language, you will see the associated filename download from GitHub, pass that filename in the next command below..`
- `udmodel_english <- udpipe_load_model(file = 'english-ud-2.0-170801.udpipe’)`
- `#Now annotate your corpus or sentence (or haiku)`
- `s <- udpipe_annotate(udmodel_english,` “An old silent pond... A frog jumps into the pond, splash! Silence again.”`)`
- `x <- data.frame(s)`
- `colnames(x)`



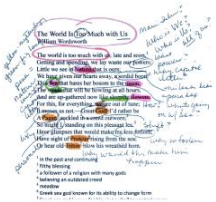
Annotating (continued)

```
> colnames(x)
 [1] "doc_id"        "paragraph_id"  "sentence_id"
 [4] "sentence"      "token_id"      "token"
 [7] "lemma"         "upos"          "xpos"
[10] "feats"         "head_token_id" "dep_rel"
[13] "deps"          "misc"
```

```
> x$token
 [1] "An"      "old"     "silent"  "pond"    "..."     "A"
 [7] "frog"    "jumps"   "into"    "the"     "pond"    ","
[13] "splash"  "!"       "Silence" "again"   "."
```

- **And your Universal Parts of Speech (UPOS):**

```
> x$upos
 [1] "DET"   "ADJ"   "ADJ"   "NOUN"  "PUNCT" "DET"   "NOUN"
 [8] "VERB"  "ADP"   "DET"   "NOUN"  "PUNCT" "NOUN"  "PUNCT"
[15] "ADV"   "ADV"   "PUNCT"
```



Getting part of speech

- `verbs <- subset(x, upos %in% c("VERB"))`
- `stats$token`

And now..

- **You can do a much better text analysis since you have tokens *and roles* in the text..**

Greek

- `udmodel <- udpipe_download_model(language = "greek")`
- `udmodel_greek <- udpipe_load_model(file = 'greek-ud-2.0-170801.udpipe’)`
- `s <- udpipe_annotate(udmodel_greek,` "Πενθώ τόν ήλιο καί πενθώ τα χρόνια που έρχονται. Χωρίς εμάς καί τραγουδώ τ’ άλλα πού πέρασαν. Εάν είναι αλήθεια. Μιλημένα τα σώματα καί οί βάρκες πού έκρουζαν γλυκά.. Οί κιθάρες πού αναβόσβησαν κάτω από τα νερά")
- `x <- data.frame(s)`
- `colnames(x)`
- `x$token`
- `x$upos`
- `verbs <- subset(x, upos %in% c("VERB"))`
- `verbs$token`

Hindi

```
> model <- udpipe_download_model(language = "hindi")
Downloading udpipe model from https://raw.githubusercontent.com/jwijf
fels/udpipe.models.ud.2.0/master/inst/udpipe-ud-2.0-170801/hindi-ud-2
.0-170801.udpipe to D:/user/docs/NU/_Info6101/Lecture 2/labs/udpipe/m
odels/hindi-ud-2.0-170801.udpipe
trying URL 'https://raw.githubusercontent.com/jwijffels/udpipe.models
.ud.2.0/master/inst/udpipe-ud-2.0-170801/hindi-ud-2.0-170801.udpipe'
Content type 'application/octet-stream' length 26137581 bytes (24.9 M
B)
downloaded 24.9 MB

> model <- udpipe_load_model(file = "hindi-ud-2.0-170801.udpipe")
> x <- udpipe_annotate(model, " मैं तन्हा हूँ मुझे तन्हा ही रहने दो, देखकर मेरे बहते
 आंसू, तुम अपने लहू न बहने दो, मैं आपका दीवाना हूँ, मुझे बस अपना पागल रहने दो    "
)#hindi poem
> x <- data.frame(x)
>
```

Hindi uPOS

```
> x$token
 [1] "मैं"      "तन्हा"    "हूँ"      "मुझे"     "तन्हा"    "ही"       "रहने"
 [8] "दो"       ","        "देखकर"   "मेरे"     "बहते"     "आंसू"     ","
[15] "तुम"      "अपने"     "लहू"      "न"        "बहने"     "दो"       ","
[22] "मैं"      "आपका"     "दीवाना"  "हूँ"      ","        "मुझे"     "बस"
[29] "अपना"     "पागल"     "रहने"     "दो"

> x$upos
 [1] "PRON"  "VERB"  "AUX"   "PRON"  "NOUN"  "PART"  "VERB"
 [8] "NUM"   "PUNCT" "VERB"  "PRON"  "VERB"  "NOUN"  "PUNCT"
[15] "NOUN"  "PRON"  "ADV"   "PART"  "VERB"  "NUM"   "PUNCT"
[22] "PRON"  "PRON"  "ADJ"   "NOUN"  "PUNCT" "PRON"  "PART"
[29] "PRON"  "ADJ"   "VERB"  "NUM"
```

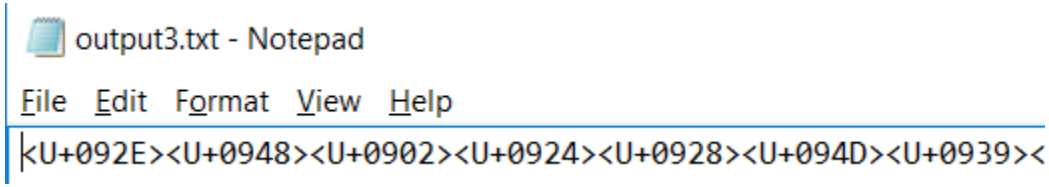
Printing Unicode to console

- `install.packages("utf8")`
- `library(utf8)`
- `utf8_print(unlist(x$token))`
- `#concatenating:`
  `paste( unlist(x$token), collapse='')`

```
> unlist(x$token)
 [1] "मैं"      "तन्हा"    "हूँ"      "मुझे"     "तन्हा"    "ही"       "रहने"
 [8] "दो"       ","        "देखकर"   "मेरे"     "बहते"     "आंसू"     ","
[15] "तुम"      "अपने"     "लहू"      "न"        "बहने"     "दो"       ","
[22] "मैं"      "आपका"    "दीवाना"   "हूँ"      ","        "मुझे"     "बस"
[29] "अपना"     "पागल"     "रहने"     "दो"
> utf8_print(unlist(x$token))
 [1] "मैं"      "तन्हा"    "हूँ"      "मुझे"     "तन्हा"    "ही"       "रहने"
 [8] "दो"       ","        "देखकर"   "मेरे"     "बहते"     "आंसू"     ","
[15] "तुम"      "अपने"     "लहू"      "न"        "बहने"     "दो"       ","
[22] "मैं"      "आपका"    "दीवाना"   "हूँ"      ","        "मुझे"     "बस"
[29] "अपना"     "पागल"     "रहने"     "दो"
> paste( unlist(x$token), collapse='')
[1] "मैंतन्हाहूँमुझेतन्हाहीरहनेदो,देखकरमेरेबहतेआंसू,तुमअपनेलहूनबहनेदो,मैंआपकादीवानाहूँ,मुझेबसअप
नापागलरहनेदो"
```

(failed attempts)

- `fileConn <- file("output3.txt",encoding="UTF-8")`
- `writeLines(paste( unlist(x$token), collapse=''), con = fileConn, sep = "\n", useBytes = FALSE)`
- `Close(fileConn)`



- `utf8ToInt(paste( unlist(x$token), collapse='’)) #prints the Unicode integer, then can use a Devanagari program to output Devanagari`
- `#Also: capture.output(anova_test, file = "tests.txt", append = TRUE)`
- `#Also: write.table(x, "x.txt", append = FALSE, sep = " ", dec = ".", row.names = TRUE, col.names = TRUE)`

Character encodings

- https://docs.python.org/2.4/lib/standard-encodings.html

Printing Hindi Unicode to file

- `writeLines(text = paste( unlist(x$token), collapse=''), con = "hindi.txt", useBytes = T)`

hindi.txt - Notepad

File Edit Format View Help

मैंतन्हाहूँमुझेतन्हाहीरहनेदो,देखकरमेरेबहतेआंसू,तुमअपनेलहूनबहनेदो,मैंआपकादीवानाहूँ,मुझेबसअपनापागलरहनेदो

Reading Hindi Unicode from file

- `hindi <- readLines(con <- file("hindi-poem.txt", encoding = "UCS-2LE"))`
  - Other option: `hindi <- readLines(con <- file("hindi-poem.txt", encoding = "UTF-16")) )`
- `close(con)`
- `unique(Encoding(hindi))`
- `x <- udpipe_annotate(model, hindi)`
- `x <- data.frame(x)`

```
> A <- readLines(con <- file("hindi-poem.txt", encoding = "UCS-2LE"))
> close(con)
> unique(Encoding(A))
[1] "UTF-8"
> A
[1] "मैं तन्हा हूँ मुझे तन्हा ही रहने दो, देखकर मेरे बहते आंसू, तुम अपने लहू न बहने दो, मैं
 आपका दीवाना हूँ, मुझे बस अपना पागल रहने दो"
> x <- udpipe_annotate(model, A)
> x <- data.frame(x)
> x$token
 [1] "मैं"      "तन्हा"    "हूँ"      "मुझे"     "तन्हा"    "ही"       "रहने"
 [8] "दो"       ","        "देखकर"   "मेरे"     "बहते"     "आंसू"     ","
[15] "तुम"      "अपने"     "लहू"      "न"        "बहने"     "दो"       ","
[22] "मैं"      "आपका"     "दीवाना"  "हूँ"      ","        "मुझे"     "बस"
[29] "अपना"     "पागल"     "रहने"     "दो"
```

References

- https://www.rdocumentation.org/packages/base/versions/3.5.0/topics/readLines
- https://www.twilio.com/docs/glossary/what-is-ucs-2-character-encoding

Chinese

```
> model <- udpipe_load_model(file = "chinese-ud-2.0-170801.udpipe")
> x <- udpipe_annotate(model, " 小娃撐小艇 , 偷采白蓮回 , 不解藏蹤跡 , 浮萍
一道開    ")#mandarin poem
> x <- data.frame(x)
> x$token
 [1] "小"    "娃撐" "小艇" ","    "偷采" "白"    "蓮"    "回"    ","
[10] "不"    "解"    "藏蹤" "跡"    ","    "浮萍" "一"    "道"    "開"
> x$upos
 [1] "PART"  "NOUN"  "NOUN"  "PUNCT" "VERB"  "PROPN" "PROPN"
 [8] "VERB"  "PUNCT" "ADV"   "VERB"  "VERB"  "NOUN"  "PUNCT"
[15] "PROPN" "NUM"   "NOUN"  "VERB"
>
```

- **writeLines(text = paste(unlist(x$token), collapse=''), con = "Chinese.txt", useBytes = T)**

chinese.txt - Notepad

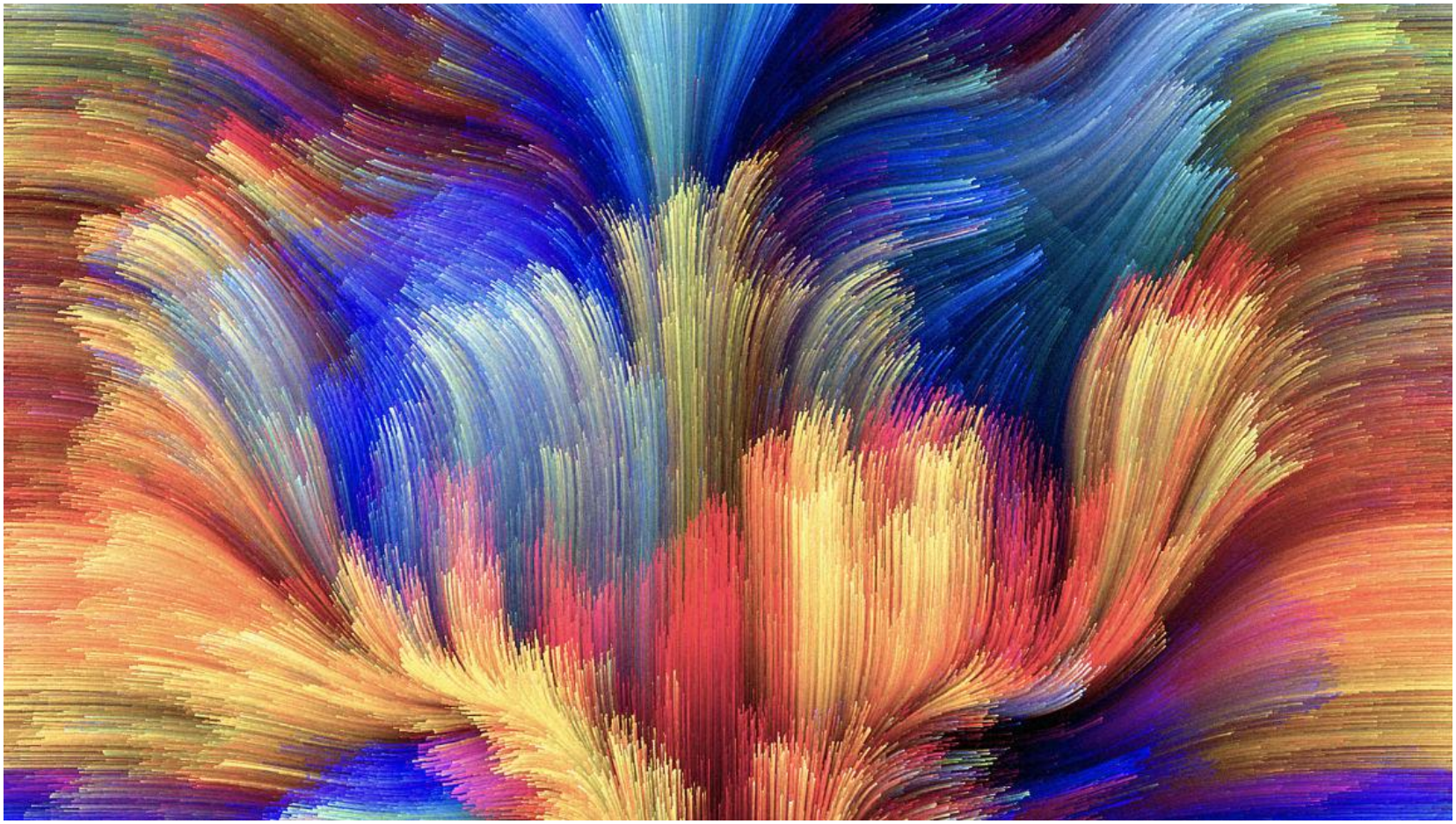
File Edit Format View Help

小娃撐小艇,偷采白蓮回,不解藏蹤跡,浮萍一道開

Challenge with Chinese

- **Written mandarin consists of ideograms (very large vocabulary)**
- **Would conversion to *pinyin* would yield a potentially more meaningful analysis?**

Power of Data!



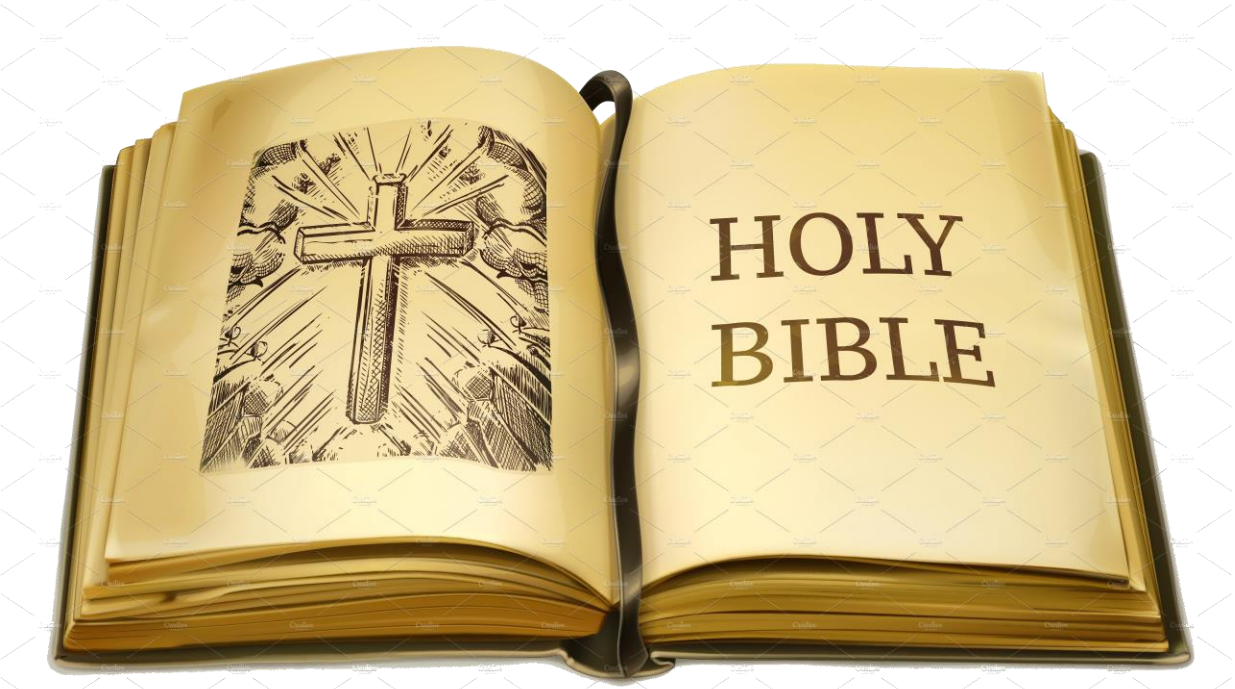
https://www.fastcompany.com/3040671/the-power-of-data-to-create-powerful-change

Most read magazine in the world?

- https://en.wikipedia.org/wiki/The_Watchtower

Most read text?

- The Christian Bible
- https://www.bible.com
- Khmer, anyone?
  - https://www.bible.com/bible/315/jhn.1.kcb



Additional References

- ***Readr***: **https://readr.tidyverse.org/articles/readr.html**
- ***Tidyverse***: **https://www.tidyverse.org/**