Mani Sarkar

github: neomatrix369
twitter: @theNeomatrix369
blogs: https://medium.com/@neomatrix369

NLP Profiler

A simple profiler, to profile textual datasets

5th Oct 2020

NLP Zurich meetup

Presentation slides: *live*

https://bit.ly/nlp-profiler-slides

Download the PDF for clickable links in the slides

https://github.com/neomatrix369/nlp_profiler/blob/master/presentations/01-nlp-zurich-2020/README.md

About me

Mani Sarkar

More about me

Freelance Software, Data, ML Engineer

Java / JVM

Cloud / Infra / DevOps

Polyglot developer

Code quality, testing, performance, DevOps, deep affinity for AI/ML/DL/NLP, NN...

LJC, Devoxx, developer communities

Strengthening teams and helping them accelerate

JCP member, F/OSS projects: @adoptopenjdk @graalvm @truffleruby

Java Champion, Oracle Groundbreaker Ambassador,
Software Crafter, Blogger, Speaker

Agenda

About the talk

- *Introduction*
- *Main talk*
- *Demo (walk-thru)*
- *Summary*
- *Resources*
- *Closing and Q&A*
- ***Appendix** section: more good stuff for later*

Thank You!

- **Kornelia** and **team**, for organising this session, and giving me a opportunity to present at this meetup
- And to “**you**”, for sparing your valuable time and trusting me

Disclaimer

- ***YMMV***
- Might have rough edges and **inaccuracies**
- Sharing our **learnings** over the past years
- Gathered ideas from **different sources**
- **Sharing ideas** and **experiences**
- The solutions discussed are not **silver bullets**

Citation

The respective authors and creators are, and remain the true owners of the images and other artifacts used in this presentation.

Thank you for your creations!

Introduction

What is profiling?

Says, Wikipedia

Data profiling is the process of **examining** the data available from an existing information source (e.g. a database or a file) and **collecting statistics** or **informative summaries** about that data.[1]

Quality checks?

Descriptive statistics?

Wikipedia: *https://en.wikipedia.org/wiki/Data_profiling*

Why is profiling important?

Look for **features**

Get the “big picture” **(high-level)**

Understand **biases** (if any)

To ***know our data*** better

Explainability

Extract **insights**

Get a “closer-look” **(granular)**

Verify and **Validate**

Find out the **traits** & **characteristics**

Assess **quality** and strength

And other reasons...

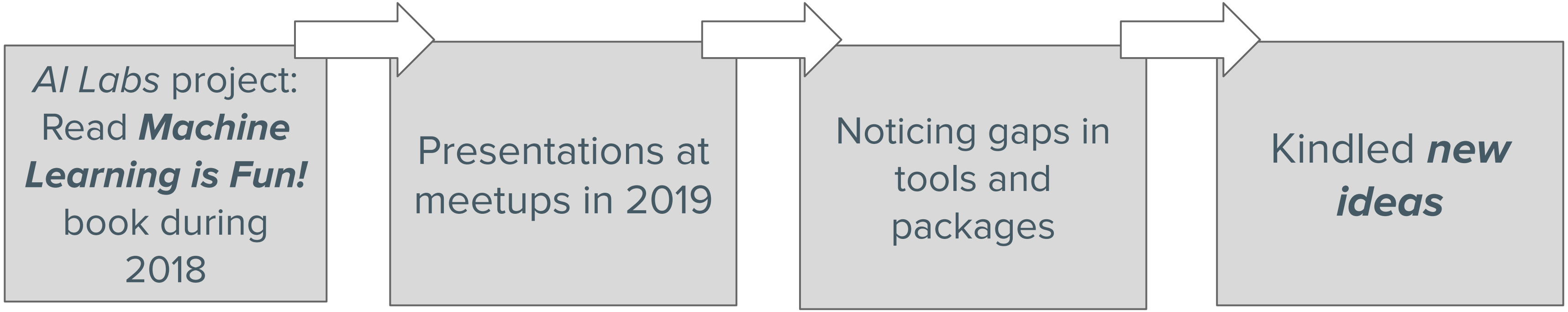
What is NLP Profiler?

- Simple python library to analyse text in your dataset
- Liken to `pandas-profiling` but works on text datasets and simple to use
- Emulates Pandas’ `describe()` function but for text datasets
- Get *microscopic (granular)* as well as *bird’s eye-view (high-level)* of your textual data
- Get *descriptive statistics* about your text
- Free/Open Source and extendable

This, is your main reason for being here...

History

How did it all get started?



AI Labs project: Read ***Machine Learning is Fun!*** book during 2018

Presentations at meetups in 2019

Noticing gaps in tools and packages

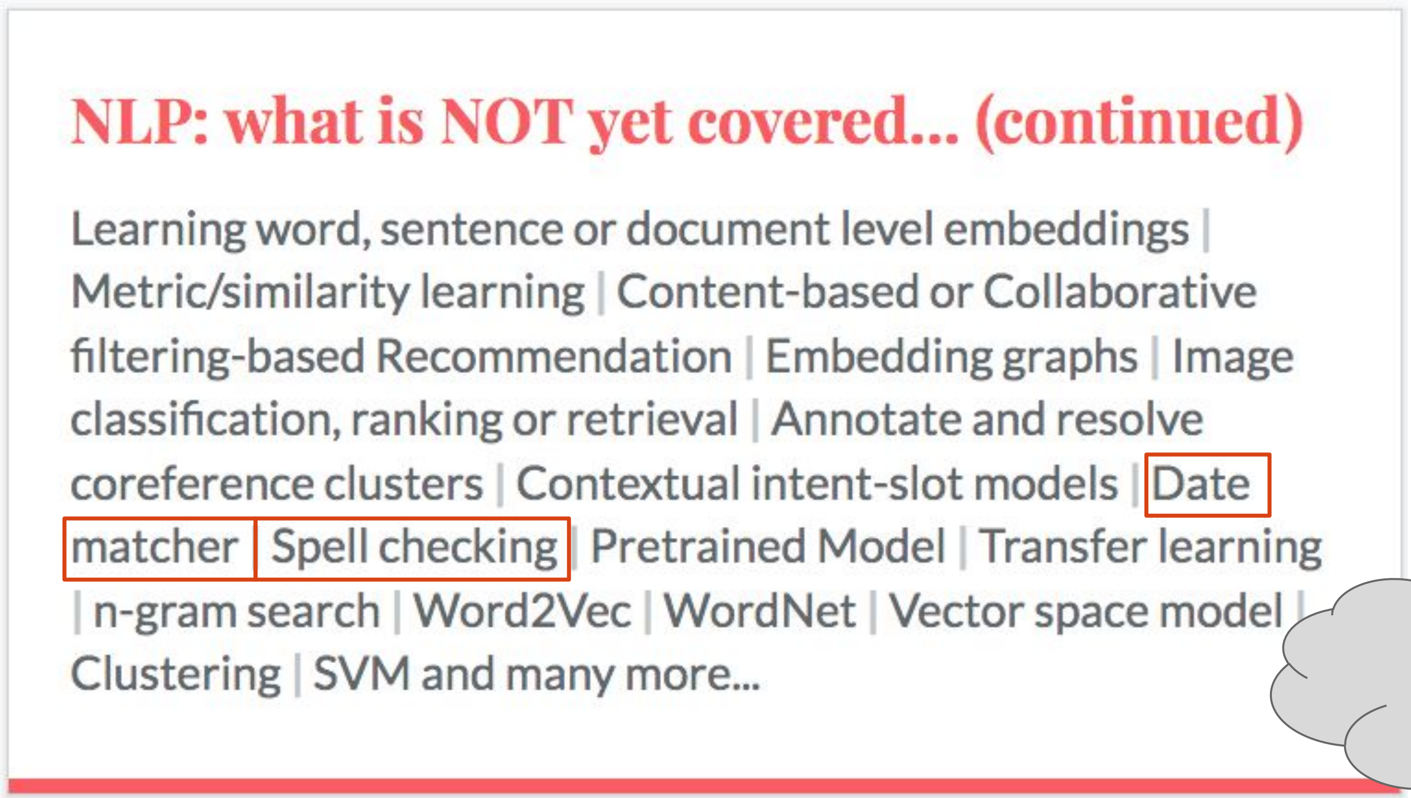
Kindled ***new ideas***

To learn more about the AI Labs initiative,

see Appendix section

How did it all get started?

Better NLP 2.0 presentation



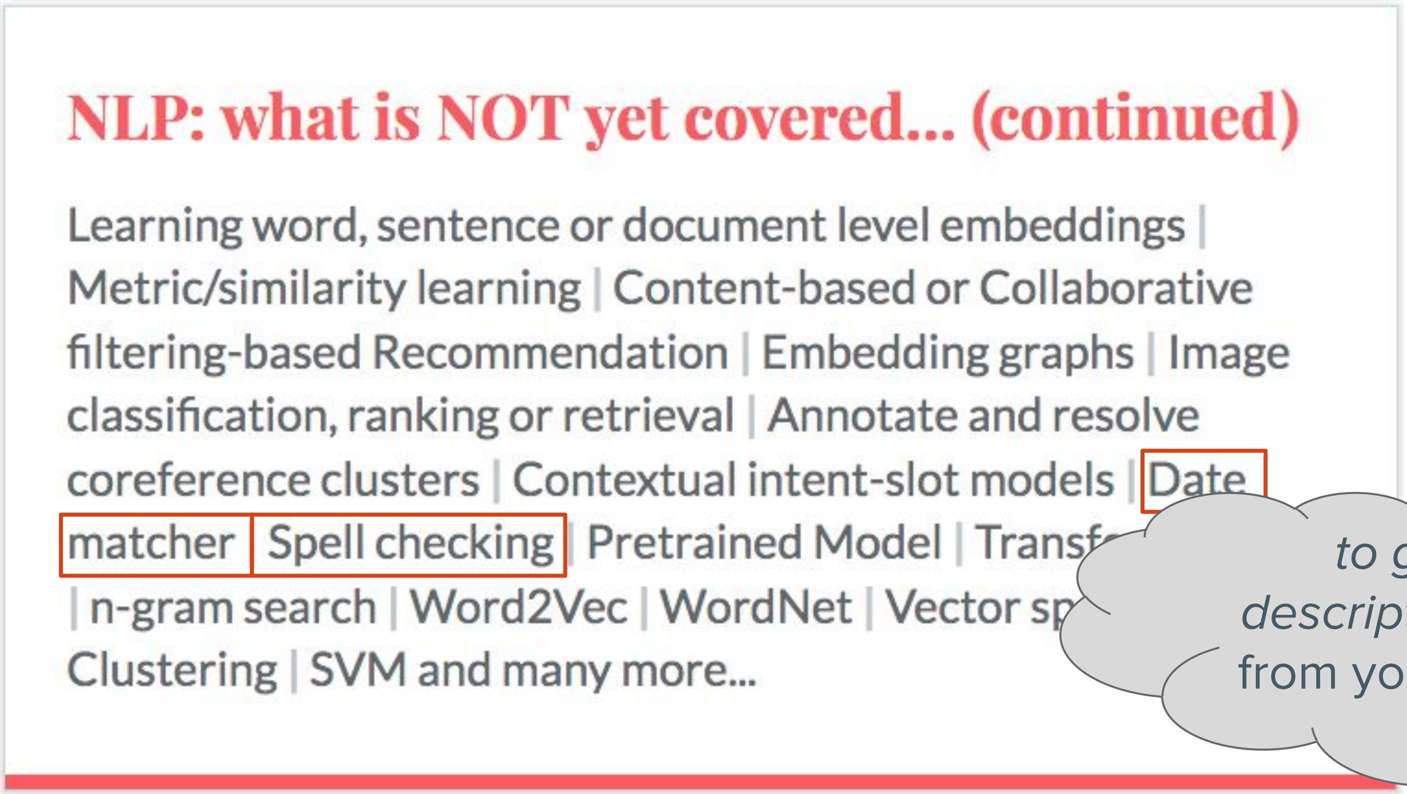
NLP: what is NOT yet covered... (continued)

Learning word, sentence or document level embeddings | Metric/similarity learning | Content-based or Collaborative filtering-based Recommendation | Embedding graphs | Image classification, ranking or retrieval | Annotate and resolve coreference clusters | Contextual intent-slot models | Date matcher | Spell checking | Pretrained Model | Transfer learning | n-gram search | Word2Vec | WordNet | Vector space model | Clustering | SVM and many more...

The library does not cover yet...

How did it all get started?

The *point* being here is we are missing tools



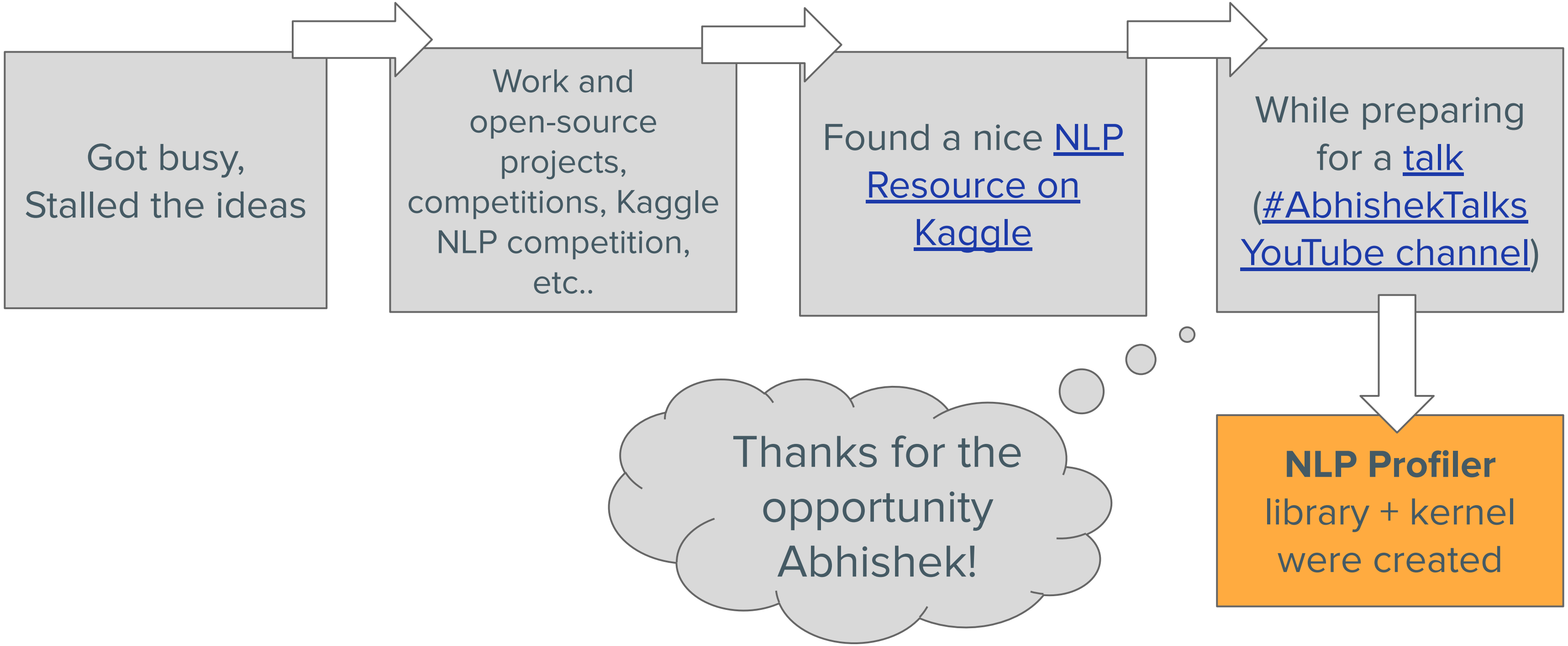
NLP: what is NOT yet covered... (continued)

Learning word, sentence or document level embeddings | Metric/similarity learning | Content-based or Collaborative filtering-based Recommendation | Embedding graphs | Image classification, ranking or retrieval | Annotate and resolve coreference clusters | Contextual intent-slot models | Date matcher | Spell checking | Pretrained Model | Transf | n-gram search | Word2Vec | WordNet | Vector sp Clustering | SVM and many more...

to generate descriptive statistics from your text data...

Thanks to Professor Ajit Jaokar for his meetups and the AI Labs initiative in London, UK during 2018 and 2019

History: how did it all get started?



Got busy, Stalled the ideas

Work and open-source projects, competitions, Kaggle NLP competition, etc..

Found a nice NLP Resource on Kaggle

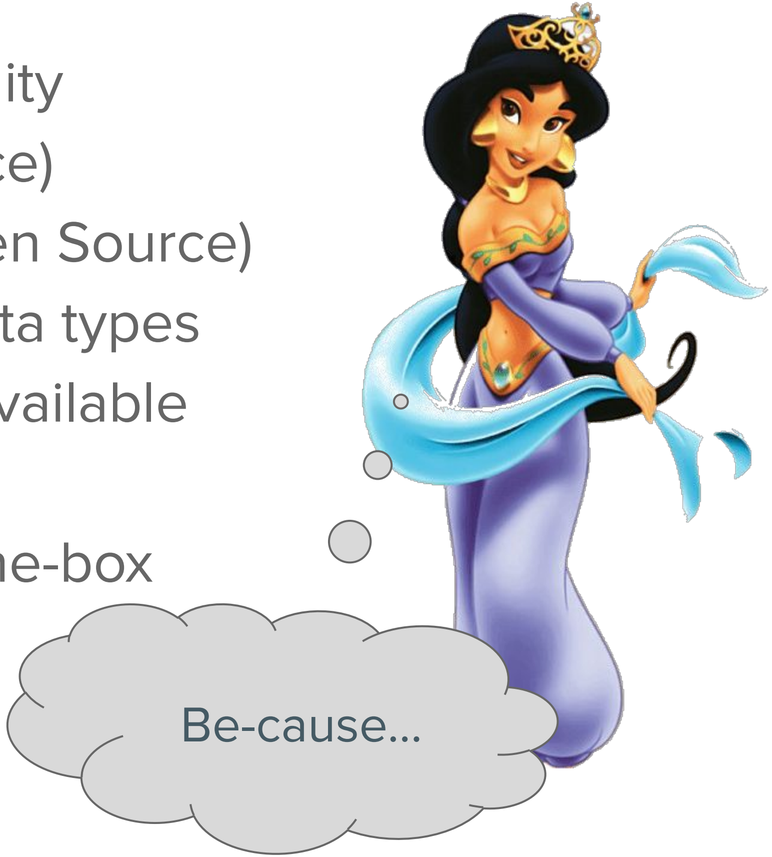
While preparing for a talk (#AbhishekTalks YouTube channel)

NLP Profiler library + kernel were created

Thanks for the opportunity Abhishek!

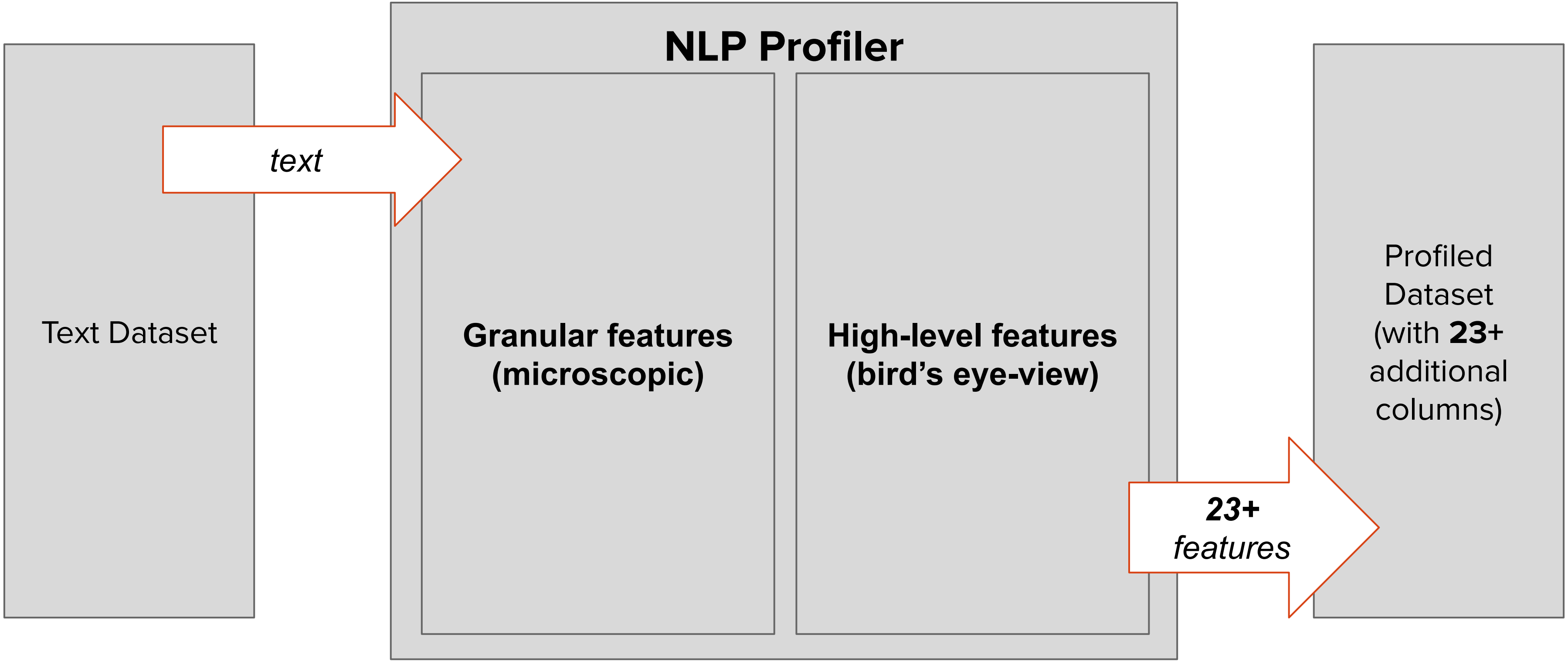
Why NLP Profiler?

- Fragmented solutions in the community
- Custom solutions (many closed-source)
- No central tool or package (Free/Open Source)
- None for text data, many for other data types
- Tools needed to create it are freely available
- Easy to put together
- Get text feature engineering out-of-the-box
- Swiss knife of tools in your toolchest



Be-cause…

How does NLP Profiler work?



Text Dataset

text

NLP Profiler

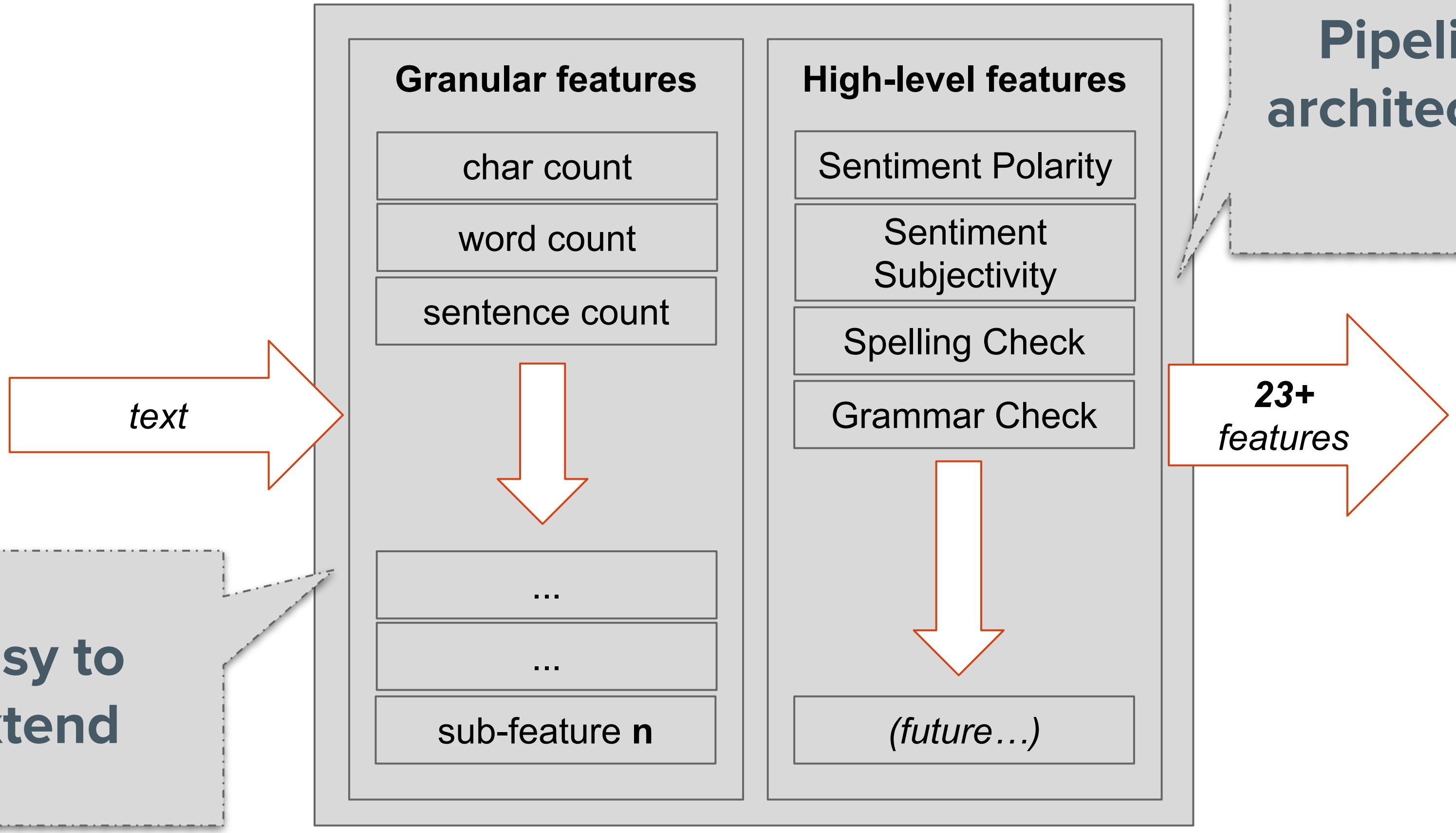
Granular features (microscopic)

High-level features (bird’s eye-view)

23+ *features*

Profiled Dataset (with **23+** additional columns)

How does NLP Profiler work?



text

Granular features

char count

word count

sentence count

...

...

sub-feature **n**

High-level features

Sentiment Polarity

Sentiment Subjectivity

Spelling Check

Grammar Check

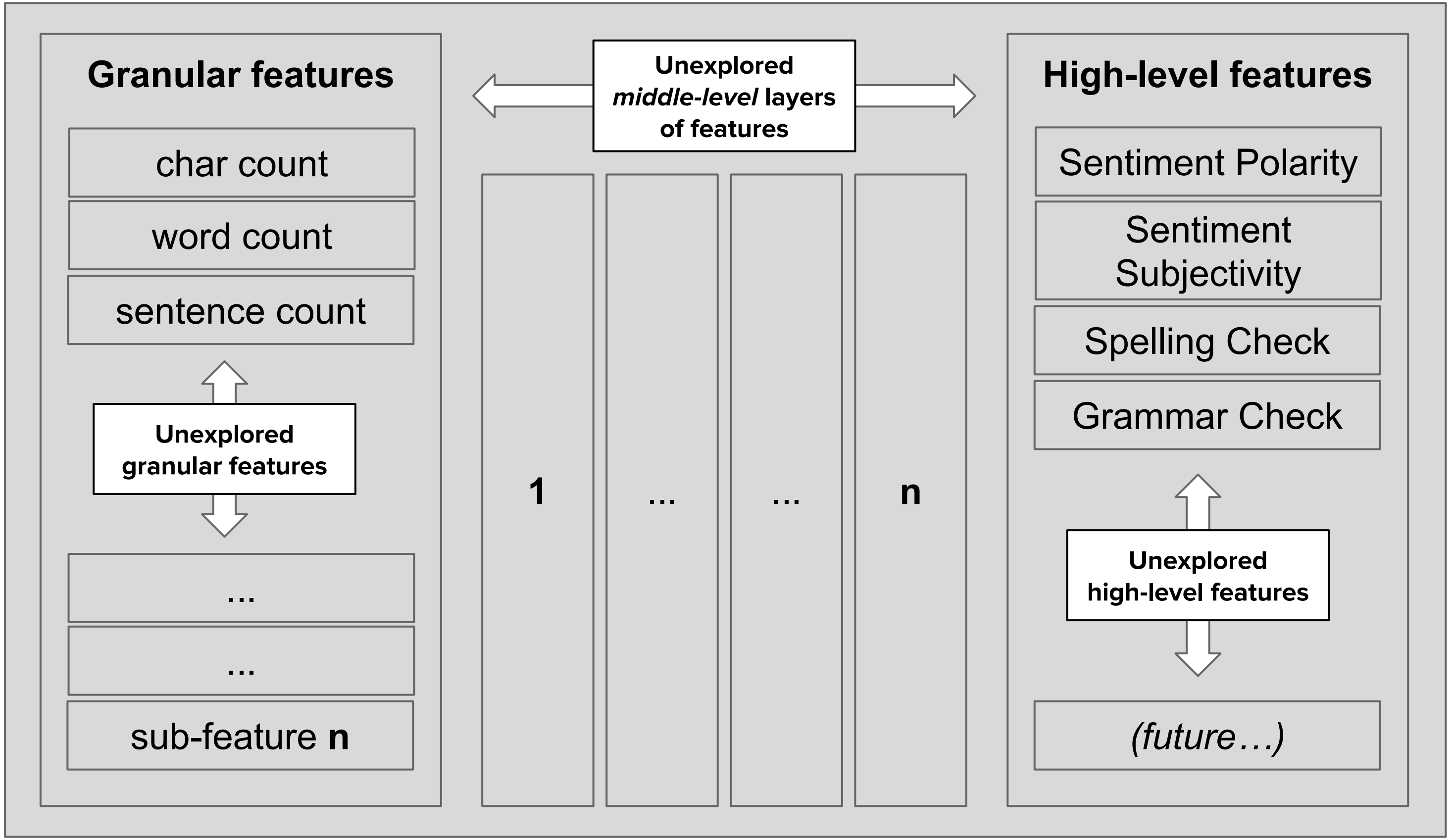
(future…)

23+ *features*

Pipeline architecture

Easy to extend

Are we complete? Gap analysis



Granular features

char count

word count

sentence count

Unexplored granular features

...

...

sub-feature **n**

Unexplored *middle-level* layers of features

1

...

...

n

High-level features

Sentiment Polarity

Sentiment Subjectivity

Spelling Check

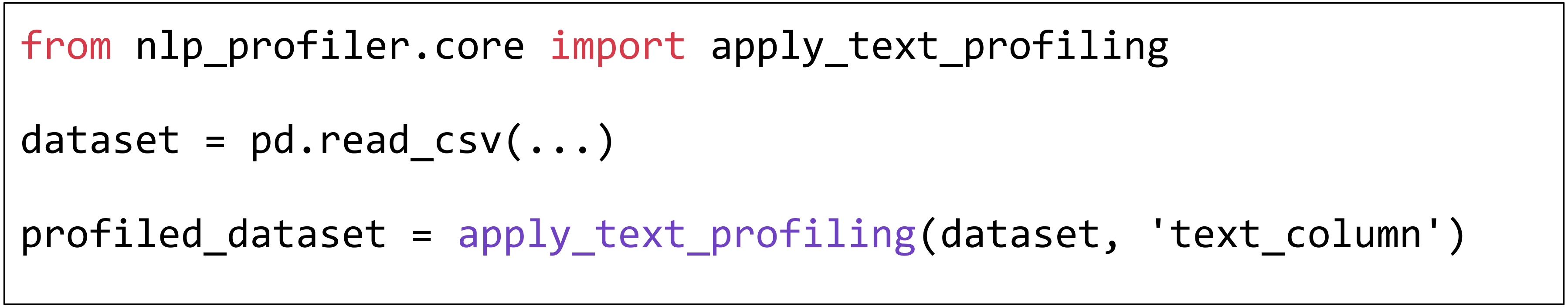
Grammar Check

Unexplored high-level features

(future…)

How to use NLP Profiler?

```
$ pip install nlp-profiler
```



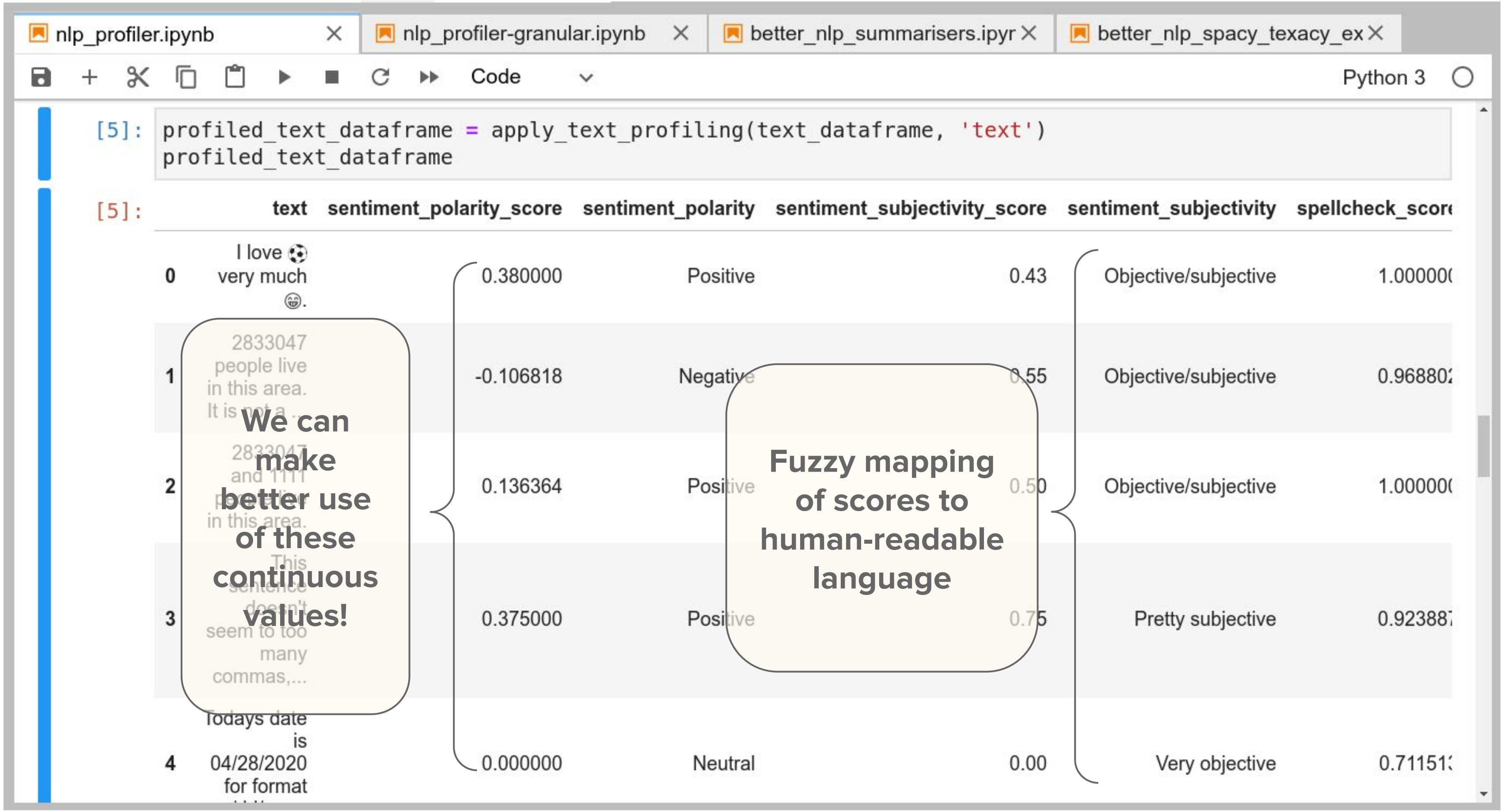
```
from nlp_profiler.core import apply_text_profiling

dataset = pd.read_csv(...)

profiled_dataset = apply_text_profiling(dataset, 'text_column')
```

https://github.com/neomatrix369/nlp_profiler#usage

Demo: walk-thru

About the demo

- **Code on GitHub:** https://github.com/neomatrix369/nlp_profiler
- **Notebook on GitHub:** **https://www.kaggle.com/neomatrix369/nlp-profiler-simple-dataset**
- Illustrates some use cases using a simple dataset
- Also shows how it can be integrated into existing workflow with widely used tools


nlp_profiler.ipynb
nlp_profiler-granular.ipynb
better_nlp_summarisers.ipyr
better_nlp_spacy_texacy_ex
Code
Python 3

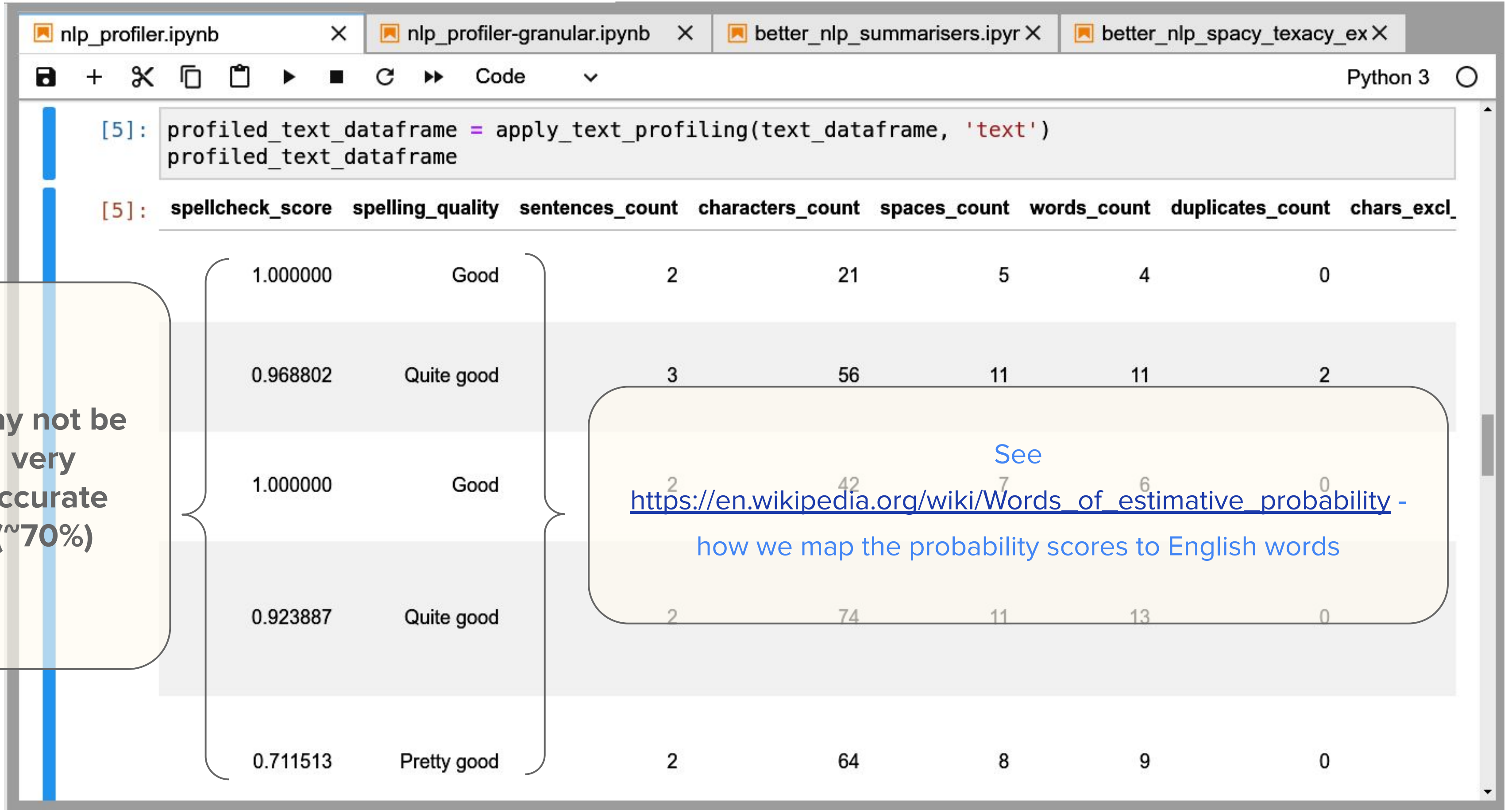
```
[5]: profiled_text_dataframe = apply_text_profiling(text_dataframe, 'text')
     profiled_text_dataframe
```

[5]:

	text	sentiment_polarity_score	sentiment_polarity	sentiment_subjectivity_score	sentiment_subjectivity	spellcheck_scor
0	I love ⚽ very much 😁.	0.380000	Positive	0.43	Objective/subjective	1.00000
1	2833047 people live in this area. It is not a ...	-0.106818	Negative	0.55	Objective/subjective	0.96880
2	2833047 and 1111 people live in this area.	0.136364	Positive	0.50	Objective/subjective	1.00000
3	This sentence doesn't seem to too many commas,...	0.375000	Positive	0.75	Pretty subjective	0.92388
4	Todays date is 04/28/2020 for format	0.000000	Neutral	0.00	Very objective	0.71151

We can make better use of these continuous values!

Fuzzy mapping of scores to human-readable language


nlp_profiler.ipynb
nlp_profiler-granular.ipynb
better_nlp_summarisers.ipyr
better_nlp_spacy_texacy_ex
Code
Python 3

```
[5]: profiled_text_dataframe = apply_text_profiling(text_dataframe, 'text')
     profiled_text_dataframe
```

[5]:

spellcheck_score	spelling_quality	sentences_count	characters_count	spaces_count	words_count	duplicates_count	chars_excl_
1.000000	Good	2	21	5	4	0	
0.968802	Quite good	3	56	11	11	2	
1.000000	Good	2	42	7	6	0	
0.923887	Quite good	2	74	11	13	0	
0.711513	Pretty good	2	64	8	9	0	

May not be very accurate (~70%)

See https://en.wikipedia.org/wiki/Words_of_estimative_probability - how we map the probability scores to English words

Word Estimative Probability

See Appendix section

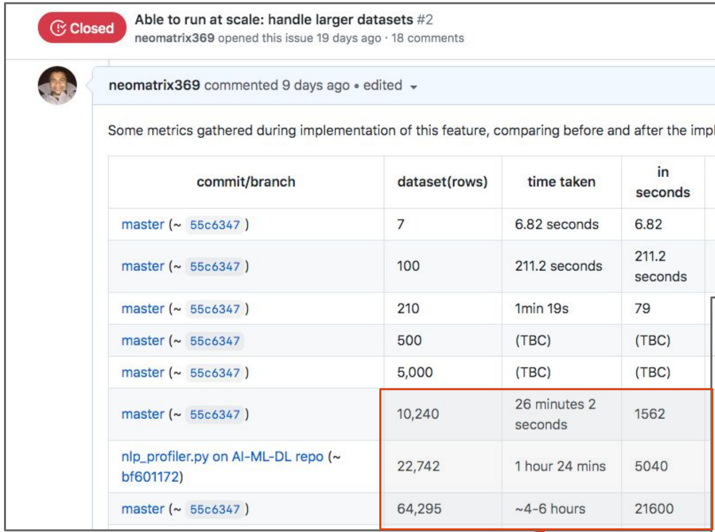
for more details

Puzzles: NLP Profiler

- What are the limitations of the NLP Profiler?
- What can we do to make it better?
- Can we make it more accurate?
- If we have scaling issues how do we tackle it?
- Any other ideas come to mind?
- What about other languages than English?

Performance Improvements

Before



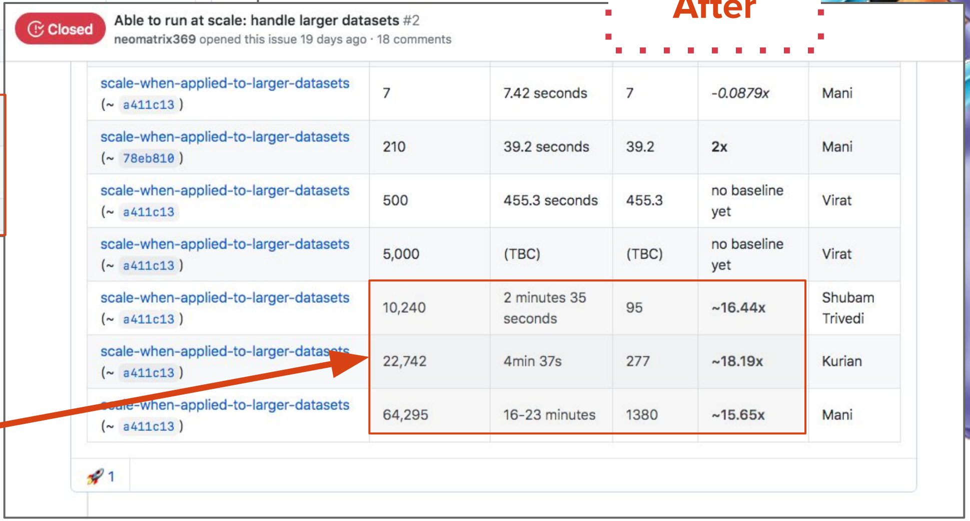
Closed **Able to run at scale: handle larger datasets** #2
neomatrix369 opened this issue 19 days ago · 18 comments

neomatrix369 commented 9 days ago • edited

Some metrics gathered during implementation of this feature, comparing before and after the implementation:

commit/branch	dataset(rows)	time taken	in seconds	speed up (x times)	run by
master (~ 55c6347)	7	6.82 seconds	6.82	baseline	Mani
master (~ 55c6347)	100	211.2 seconds	211.2 seconds	baseline	Virat
master (~ 55c6347)	210	1min 19s	79		
master (~ 55c6347	500	(TBC)	(TBC)		
master (~ 55c6347)	5,000	(TBC)	(TBC)		
master (~ 55c6347)	10,240	26 minutes 2 seconds	1562		
nlp_profiler.py on AI-ML-DL repo (~ bf601172)	22,742	1 hour 24 mins	5040		
master (~ 55c6347)	64,295	~4-6 hours	21600		

After



Closed **Able to run at scale: handle larger datasets** #2
neomatrix369 opened this issue 19 days ago · 18 comments

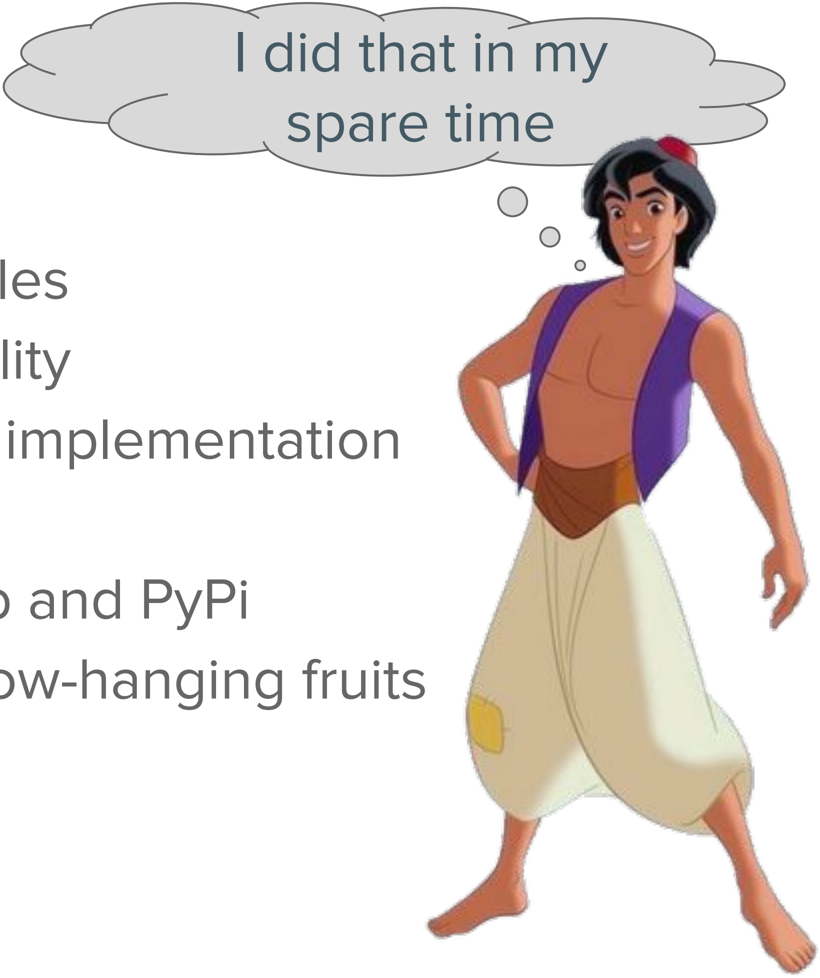
scale-when-applied-to-larger-datasets (~ a411c13)	7	7.42 seconds	7	*-0.0879x*	Mani
scale-when-applied-to-larger-datasets (~ 78eb810)	210	39.2 seconds	39.2	**2x**	Mani
scale-when-applied-to-larger-datasets (~ a411c13	500	455.3 seconds	455.3	no baseline yet	Virat
scale-when-applied-to-larger-datasets (~ a411c13)	5,000	(TBC)	(TBC)	no baseline yet	Virat
scale-when-applied-to-larger-datasets (~ a411c13)	10,240	2 minutes 35 seconds	95	**~16.44x**	Shubam Trivedi
scale-when-applied-to-larger-datasets (~ a411c13)	22,742	4min 37s	277	**~18.19x**	Kurian
scale-when-applied-to-larger-datasets (~ a411c13)	64,295	16-23 minutes	1380	**~15.65x**	Mani

🚀 1

15x to 18x speed-up
(link to stats)

Infrastructure works

- Refactoring into cohesive modules
- Formatting the code for readability
- Retrofitting tests across original implementation
- Improving test coverage
- Shell scripts to upload to GitHub and PyPi
- Docs, references and all other low-hanging fruits



Notebooks / kernels

Check these out, different examples...

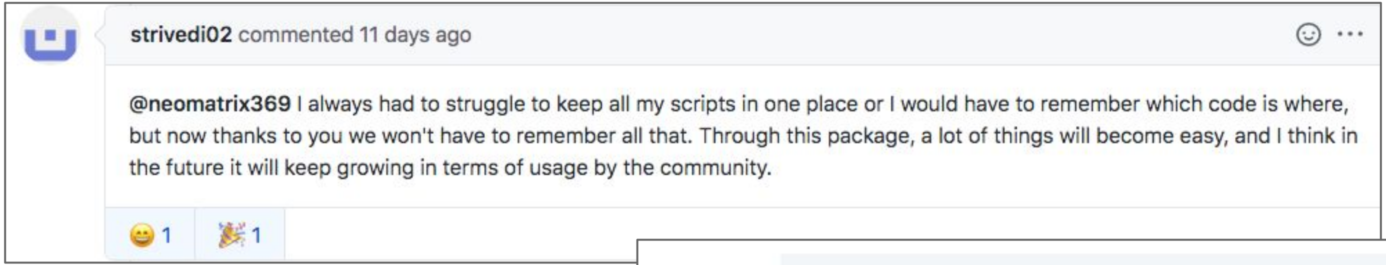
- NLP Profiler: simple dataset
- CTDS: answering the "what..." question differently
- ChaiEDA: Google Play Store Apps - review analysis
- Google Colab / Jupyter Notebooks on the Git Repo
- See notebooks/kernels from our supporters

Future plans

Together we can create this future

- More granular and high-level features
- Investigate *middle-level* features
- Ability to add your custom features while profiling
- Support multiple written languages not just English
- R language version of the library in the making
- Performance tune other aspects of the library
- Make more examples available
- And many more...

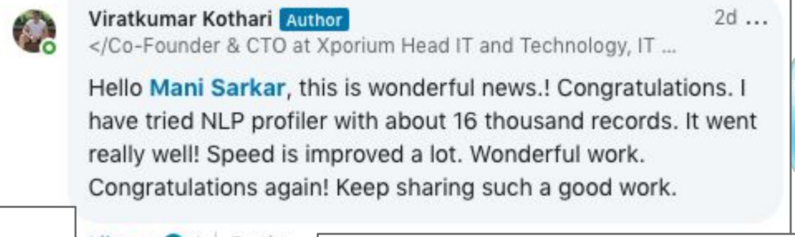
What others are saying after using it?



strivedi02 commented 11 days ago

@neomatrix369 I always had to struggle to keep all my scripts in one place or I would have to remember which code is where, but now thanks to you we won't have to remember all that. Through this package, a lot of things will become easy, and I think in the future it will keep growing in terms of usage by the community.

😄 1 🎉 1



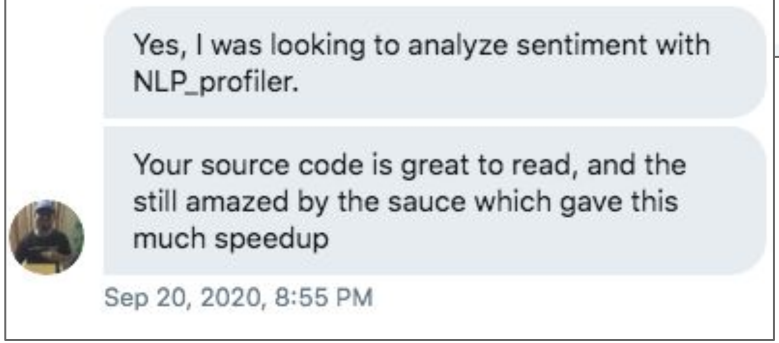
Viratkumar Kothari Author 2d ...

</Co-Founder & CTO at Xporium Head IT and Technology, IT ...

Hello **Mani Sarkar**, this is wonderful news.! Congratulations. I have tried NLP profiler with about 16 thousand records. It went really well! Speed is improved a lot. Wonderful work. Congratulations again! Keep sharing such a good work.

Like · 👍 1 | Reply

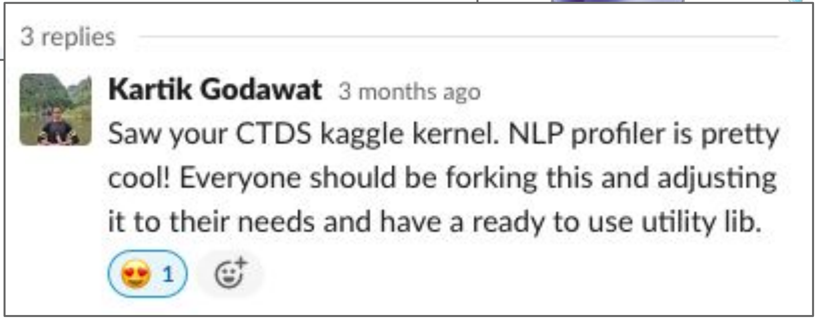
Credits and supporters page



Yes, I was looking to analyze sentiment with NLP_profiler.

Your source code is great to read, and the still amazed by the sauce which gave this much speedup

Sep 20, 2020, 8:55 PM



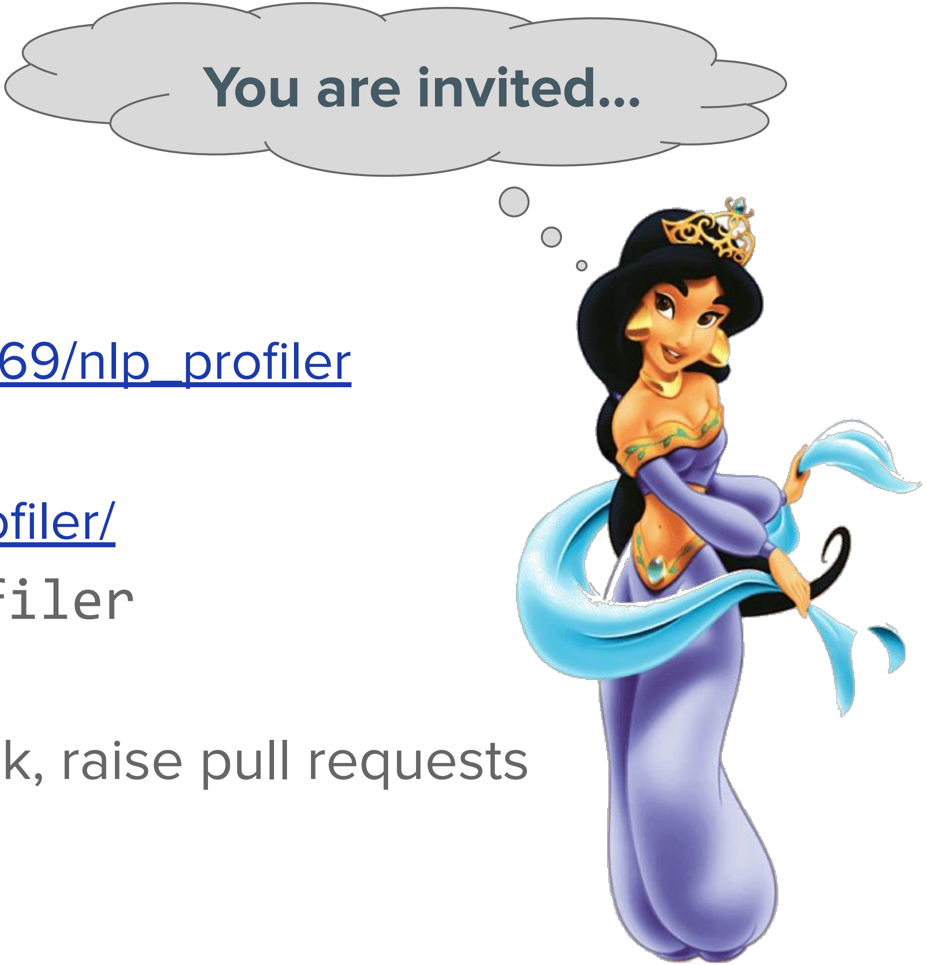
3 replies

Kartik Godawat 3 months ago

Saw your CTDS kaggle kernel. NLP profiler is pretty cool! Everyone should be forking this and adjusting it to their needs and have a ready to use utility lib.

😍 1

Get involved



- GitHub repo
  - https://github.com/neomatrix369/nlp_profiler
- On PyPi
  - https://pypi.org/project/nlp-profiler/
- Install: `pip install nlp-profiler`
- Please given it a whirl
- And share constructive feedback, raise pull requests

Summary

In summary

- One central place to find your NLP recipes
- Free/Open Source package
- Extendable and customisable
- Add/extend your existing toolkit
- At the moment can only process English language
- Lots of resources and help available
- Growing usage and community
- A *Swiss knife* among other NLP tools in the tool chest

Resources

General

- More about me
- My thoughts on many things AI/ML/DL/NLP
- AI/ML/DL resources
- NLP Zurich Meetup
- NLP Zurich Meetup NLP Profiler Event page
- NLP Zurich on LinkedIn
- NLP Zurich YouTube channel
- Email: nlp.zurich@gmail.com

NLP Specific

- NLP Profiler on Github
- NLP Profiler on PyPi
- Better NLP library
- NLP resources on Awesome AI/ML/DL repo
- Notebooks/Kernels
    - NLP Profiler: simple dataset
    - CTDS: answering the "what..." question differently
    - ChaiEDA: Google Play Store Apps - review analysis
    - Google Colab / Jupyter Notebooks on the Git Repo
    - See notebooks/kernels from our supporters
- How we map the probability scores to English words? (Words of Estimative probability)

Closing note and Thanks

We were the mascots

It has been an honour and a pleasure to be here. *Kornelia & team* are very efficient & organised!

Call us sidekicks, if you will

Contributors & supporters

Our current contributors and supporters

Thanks to the contributors even though a small number of them. We really appreciate your efforts.

Contact and keep in touch

- twitter: @theNeomatrix369
- medium: https://medium.com/@neomatrix369
- github: https://github.com/neomatrix369/
- linkedin: https://www.linkedin.com/in/mani-sarkar/
- youtube: channel | playlists
- about me: https://neomatrix369.wordpress.com/about

Q & A

How can I use .... to ...?

What do I need to do to... using **NLP Profiler?**

How can we make it better?

Jas = Jasmine? From the movie "Aladdin and ..."?

How to share recipes, ideas, patterns?

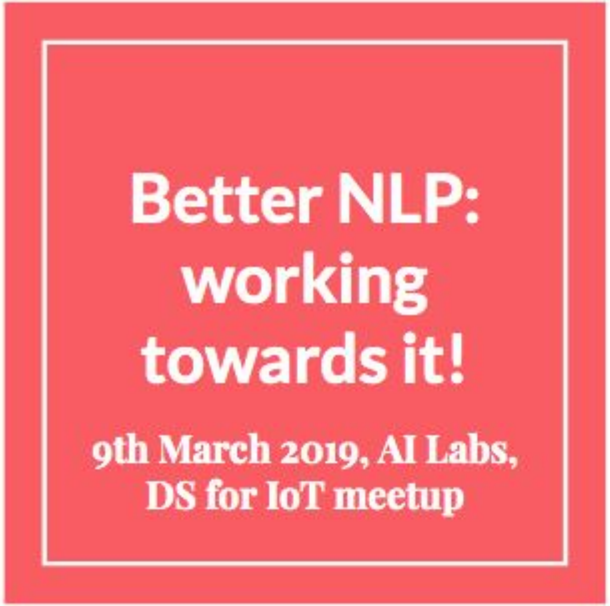
How do we contribute back?

Being a fictional star: Jas may not be able to answer questions! 😉

Appendix

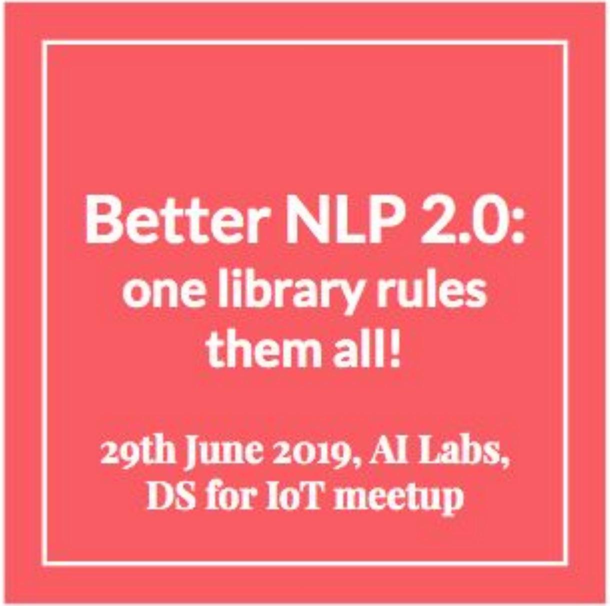
AI Labs initiative

How did it all get started?



Better NLP: working towards it!

9th March 2019, AI Labs, DS for IoT meetup



Better NLP 2.0: one library rules them all!

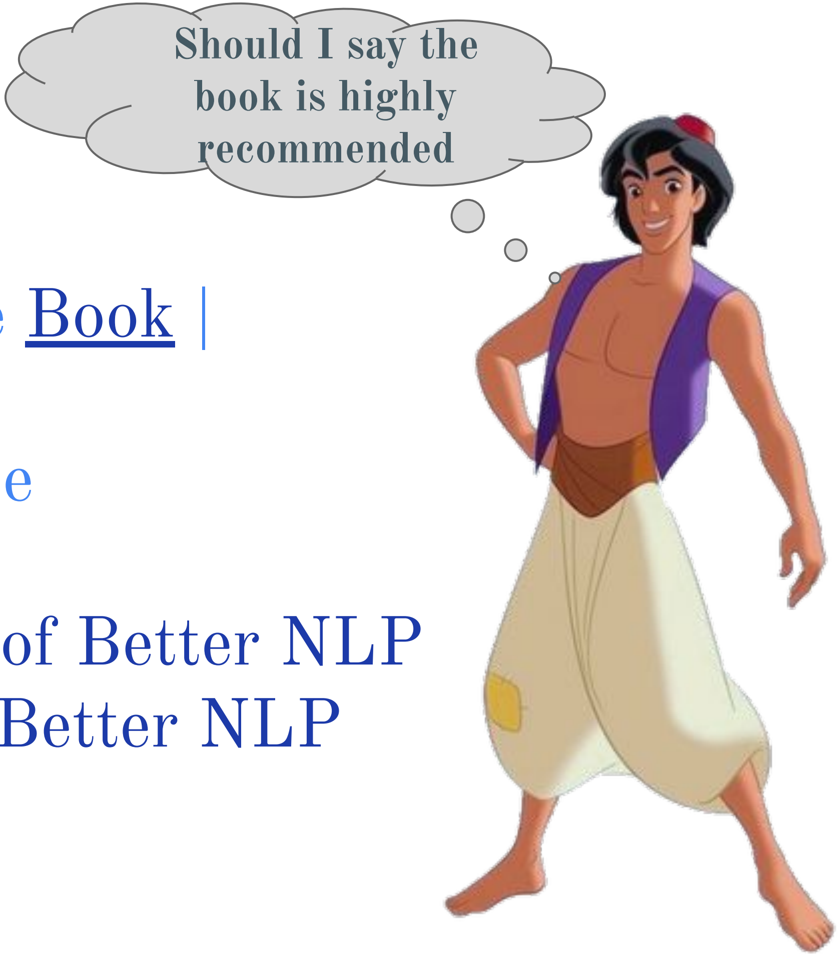
29th June 2019, AI Labs, DS for IoT meetup

http://bit.ly/better-nlp-launch

(look inside the folder ***presentations****)*

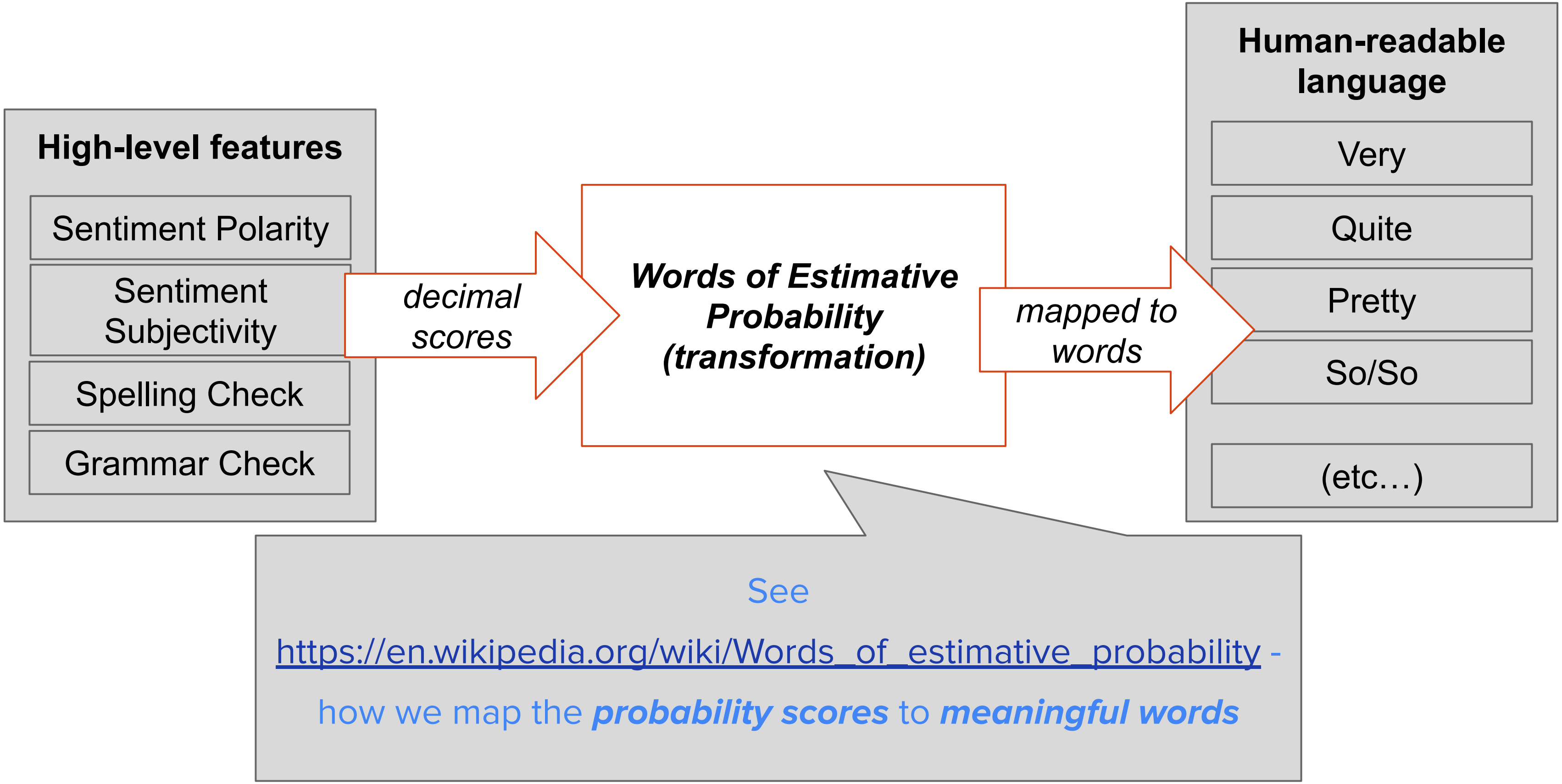
Meetups and AI Labs

- Machine Learning is Fun! See Book | Tutorial / Blogs
- Better NLP library launch, see presentations:
    - First presentation: launch of Better NLP
    - Follow-up presentation of Better NLP



Word Estimative Probability

How does NLP Profiler work?



High-level features

Sentiment Polarity

Sentiment Subjectivity

Spelling Check

Grammar Check

decimal scores

Words of Estimative Probability (transformation)

mapped to words

Human-readable language

Very

Quite

Pretty

So/So

(etc…)

See https://en.wikipedia.org/wiki/Words_of_estimative_probability - how we map the ***probability scores*** to ***meaningful words***

Word Estimative Probability

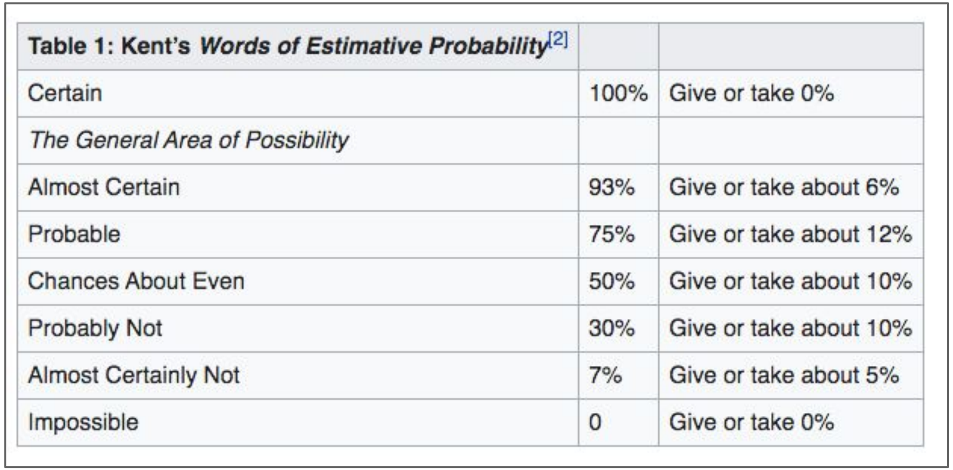
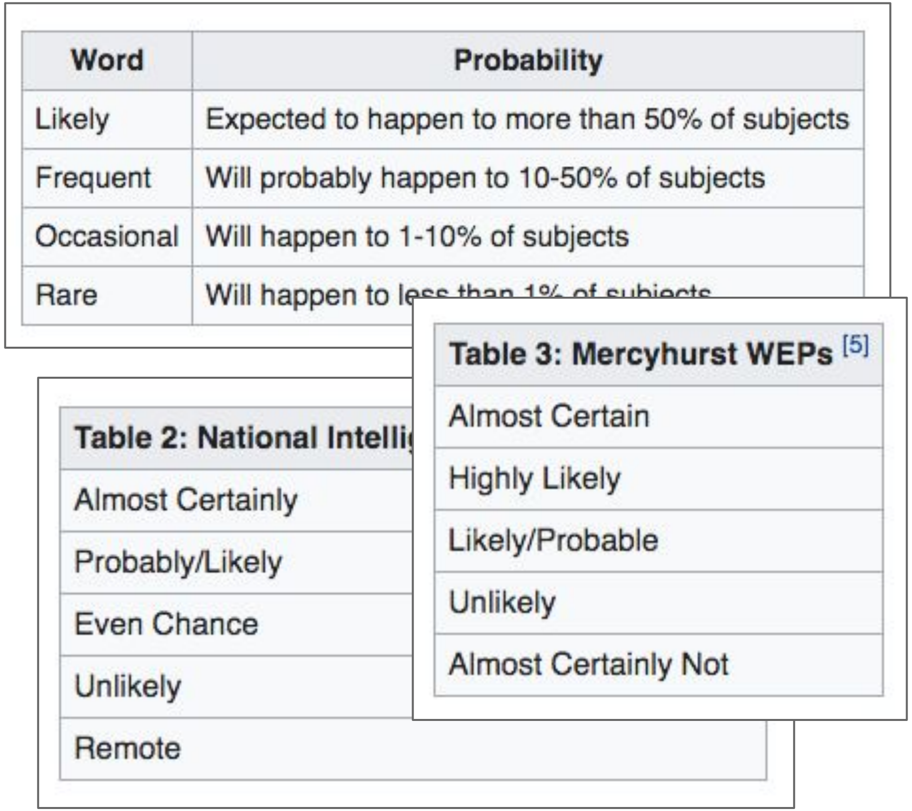


Table 1: Kent's *Words of Estimative Probability*[2]		
Certain	100%	Give or take 0%
The General Area of Possibility		
Almost Certain	93%	Give or take about 6%
Probable	75%	Give or take about 12%
Chances About Even	50%	Give or take about 10%
Probably Not	30%	Give or take about 10%
Almost Certainly Not	7%	Give or take about 5%
Impossible	0	Give or take 0%

https://en.wikipedia.org/wiki/Words_of_estimative_probability

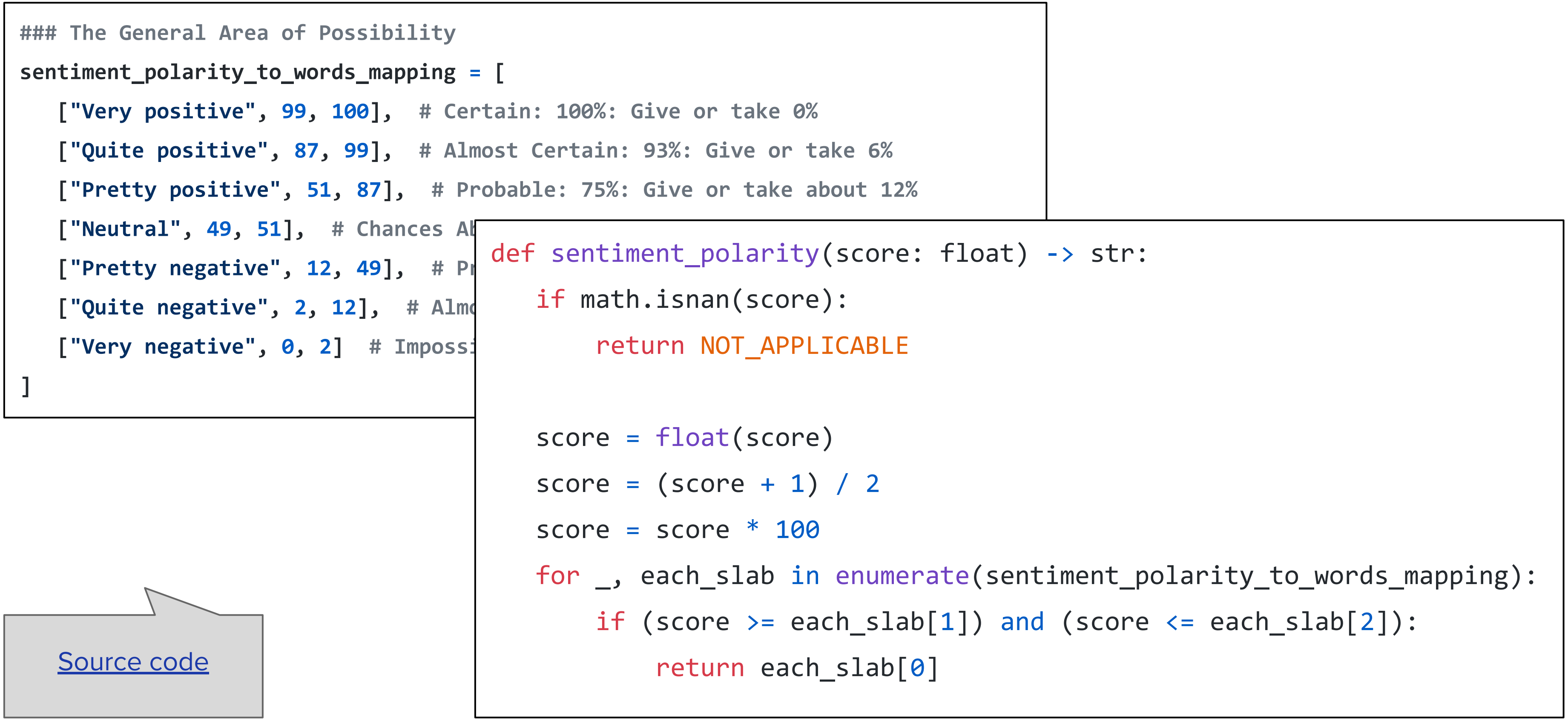


Word	Probability
Likely	Expected to happen to more than 50% of subjects
Frequent	Will probably happen to 10-50% of subjects
Occasional	Will happen to 1-10% of subjects
Rare	Will happen to less than 1% of subjects

Table 2: National Intelli
Almost Certainly
Probably/Likely
Even Chance
Unlikely
Remote

Table 3: Mercyhurst WEPs [5]
Almost Certain
Highly Likely
Likely/Probable
Unlikely
Almost Certainly Not

Word Estimative Probability (code)



```
### The General Area of Possibility
sentiment_polarity_to_words_mapping = [
   ["Very positive", 99, 100],  # Certain: 100%: Give or take 0%
   ["Quite positive", 87, 99],  # Almost Certain: 93%: Give or take 6%
   ["Pretty positive", 51, 87],  # Probable: 75%: Give or take about 12%
   ["Neutral", 49, 51],  # Chances Ab
   ["Pretty negative", 12, 49],  # Pr
   ["Quite negative", 2, 12],  # Almo
   ["Very negative", 0, 2]  # Impossi
]
```

```
def sentiment_polarity(score: float) -> str:
    if math.isnan(score):
        return NOT_APPLICABLE

    score = float(score)
    score = (score + 1) / 2
    score = score * 100
    for _, each_slab in enumerate(sentiment_polarity_to_words_mapping):
        if (score >= each_slab[1]) and (score <= each_slab[2]):
            return each_slab[0]
```

Source code

Examples

NLP examples

- Example 1
  - Github
  - Blog post
- Example 2
  - Blog post
- Example 3
  - Blog post
- Better NLP

Jupyter Notebook example

- Example 1
  - Github
  - Blog: Exploring NLP concepts using Apache OpenNLP inside a Jupyter notebook
- Example 2
  - Blog post
- Example 3
  - Github
  - Blog post

graql-to-english, english-to-graql example

- Presentation
- Github

Others

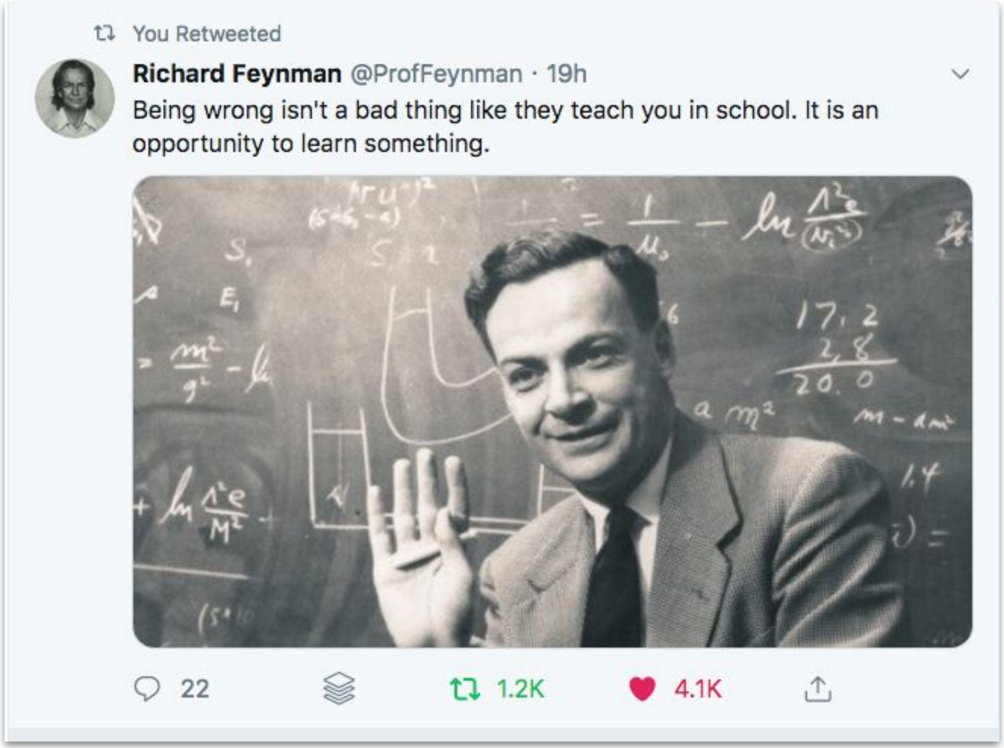
Previous talks

- I recently gave a talk: From backend development to machine learning
- “nn” things every Java developer should know about AI/ML/DL
- Naturally, getting productive, my journey with Grakn and Graql
- Do we know our data as well as our tools?
- Java N.n: What to know? How to learn?
- Some of my other talks a can be found here and here (and others on Slideshare)

One may find these methods ***unconventional*** or ***non-mainstream*** but they do work and give good results!

Being wrong isn’t bad...

Me believe same, I learn so much!



You Retweeted

Richard Feynman @ProfFeynman · 19h

Being wrong isn't a bad thing like they teach you in school. It is an opportunity to learn something.

22 1.2K 4.1K

Freebies!

Get $500 worth of free cloud credits on Oracle Cloud