State-of-the-art Natural Language Processing for 200+ Languages with 1000+ models in 1 Line of code

Pip install nlu

John Snow LABS
NLU

Afrikaans af
Arabic ar
Azeri az
Bulgarian bg
Bislama bi
Bengali bn
Breton br
Catalan ca
Czech cs
Welsh cy
Danish da
German de
Ewe ee
Greek el
English en
Esperanto eo
Spanish es
Estonian et
Basque eu
Farsi fa
Finnish fi
Fiji fj
French fr
Irish ga
Galician gl
Manx gv
Hausa ha
Hebrew he
Hindi hi
Hiri Motu ho
Haitian ht
Hungarian hu
Armenian hy
Indonesian id
Igbo ig
Icelandic is
Italian it
Japanese ja
Georgian ka
Kongo kg
Kuanyama kj
Greenlandic kl
Korean ko
Latin la
Ganda lg
Lingala ln
Luba-Katanga lu
Latvian lv
Malagasy mg
Marshallese mh
FYRO Macedonian mk
Malayalam ml
Marathi mr
Maltese mt
Ndonga ng
Dutch nl
Norwegian Bokmal no
Chichewa ny
Oromoor om
Punjabi pa
Polish pl
Portuguese pt
Kirundi rn
Romanian ro
Russian ru
Kinyarwanda rw
Sangro sg
Slovak sk
Slovenian sl
Samoan sm
Shona sn
Somali so
Albanian sq
Siswati ss
Sesotho st
Swedish sv
Thai th
Tigrinya ti
Tagalog tl
Tswana tn
Tonga to
Turkish tr
Tsonga ts
Twi tw
Tahitian ty
Ukrainian uk
Urdu ur
Venda ve
Vietnamese vi
Walloon wa
Xhosa xh
Yoruba yo
Chinese zh
Zulu zu
... 94 more!

Tackle any NLP problem in 200 languages

- Part1: Easy 1 Liners
  - Spell checking/Sentiment/POS/NER/ BERTtology embeddings
- Part2: Data analysis and NLP tasks on Bitcoin News Headline dataset
  - Preprocessing and extracting Emotions, Keywords, Named Entities and visualize them
- Part3: NLU Multi-Lingual 1 Liners with Microsoft's Marian Models
  - Translate between 200+ languages (and classify lang afterwards)
- Part 4: Data analysis and NLP tasks on **Chinese News Article** Dataset
  - Word Segmentation, Lemmatization, Extract Keywords, Named Entities and translate to english
- Part 5: Train a sentiment Classifier that **understands 100+ Languages**
  - Train on a french sentiment dataset and predict sentiment of 100+ langauges wit Multi Lingual Bert Embeddings
- Part 6: Question answering, Summarization, Squad and more with Google's T5
  - T5 Question answering and 18 + other NLP tasks (SQUAD/GLUE/SUPER GLUE)

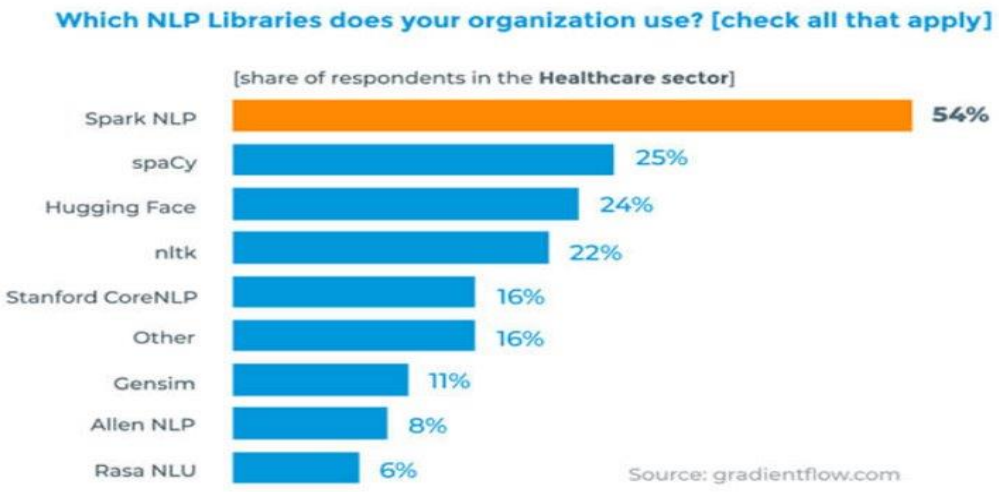
INTRODUCING SPARK NLP

State of the art NLP:

1. **Accuracy**
2. **Speed**
3. **Scalability**

Open-Source Python, Java & Scala Libraries

100+ Pre-Trained Models & Pipelines



"18-month-old Spark NLP library is already **in use by 16% of enterprise** – twice as popular as the next NLP library on the list."

AI Adoption in the Enterprise, O'Reilly, February 2019

"Spark NLP is geared towards production use in software systems that **outgrow older libraries** such as spaCy, nltk, and CoreNLP."

Wikipedia, October 2019

AI
ARTIFICIAL INTELLIGENCE
EXCELLENCE
AWARD
2020

"John Snow Labs wins our best AI product or service award thanks to **exceptional success turning AI research into real & dependable systems for a global community**."

"An open source project, tool, or contribution that **significantly advances the state of data science** is recognized with this award."

HEALTHCARE ANALYTICS
PROVIDER OF THE YEAR 2020
Healthcare Tech OUTLOOK

"By all accounts, John Snow Labs has created **the most accurate software in history** to extract facts from unstructured text."

INTRODUCING THE NLU LIBRARY

1000+ Models, 200+ Languages, 1 Line of Code

Powerful One-Liners

Over a thousand NLP models in hundreds of languages are at your fingertips with just one line of code

Elegant Python

Directly read and write pandas dataframes for frictionless integration with other libraries and existing ML pipelines

100% Open Source

Including pre-trained models & pipelines

How does it work?

```
model= nlu.load(model)
```

- Returns a nlu pipeline object

```
model.predict(data)
```

- Returns a pandas DF

How does it work?

```
model = nlu.load('emotion')
```

- Returns a nlu pipeline object

```
model.predict('I love NLU!')
```

- Returns a pandas DF

EMOTION DETECTION

```
nlu.load('emotion').predict('I love NLU!')
```

sentence_embeddings	category_sentence	category_surprise	category_sadness	category_joy	category_fear	sentence	category	id
[0.027570432052016258, -0.052647676318883896, ...]	0	0.012899903	0.0015578865	0.9760173	0.0095249	I love NLU!	joy	1

NLU WORKS DIRECTLY ON TYPICAL PYTHON DATASETS

Strings

```
import nlu
nlu.load('sentiment').predict('This is just one string')
```

Lists

```
import nlu
nlu.load('sentiment').predict(['This is an arrray', ' Of strings!'])
```

Pandas data frame

```
import nlu
import pandas as pd
data = {"text": ['This day sucks', 'I love this day', 'I dont like Sami']}
text_df = pd.DataFrame(data)
nlu.load('sentiment').predict(text_df)
```

Pandas series

```
import nlu
import pandas as pd
data = {"text": ['This day sucks', 'I love this day', 'I dont like Sami']}
text_df = pd.DataFrame(data)
nlu.load('sentiment').predict(text_df['text'])
```

Spark
Data Frame

Ray
Data Frame

Dask
Data Frame

Part of Speech for Bengali (POS)

```
# 'The village is also called 'Mod' in Tora language' in Behgali
nlu.load("bn.pos").predict("বাসস্থান-ঘরগৃহস্থালি তোড়া ভাষায় গ্রামকেও বলে ` মোদ ' ।")
```

token	pos
বাসস্থান-ঘরগৃহস্থালি	NN
তোড়া	NNP
ভাষায়	NN
গ্রামকেও	NN
বলে	VM
`	SYM
মোদ	NN
'	SYM
।	SYM

```
# 'In 1918, the forces of the Arab Revolt liberated Damascus with the help of the British' in Arabic
nlu.load('ar.ner').predict('في عام 1918 حررت قوات الثورة العربية دمشق بمساعدة من الإنكليز')
```

entity_class	ner_confidence	entities
ORG	[1.0, 1.0, 1.0, 0.9997000098228455, 0.9840999841690063, 0.9987999796867371, 0.9990000128746033, 0.9998999834060669, 0.9998999834060669, 0.9993000030517578, 0.9998999834060669]	قوات الثورة العربية
LOC	[1.0, 1.0, 1.0, 0.9997000098228455, 0.9840999841690063, 0.9987999796867371, 0.9990000128746033, 0.9998999834060669, 0.9998999834060669, 0.9993000030517578, 0.9998999834060669]	دمشق
PER	[1.0, 1.0, 1.0, 0.9997000098228455, 0.9840999841690063, 0.9987999796867371, 0.9990000128746033, 0.9998999834060669, 0.9998999834060669, 0.9993000030517578, 0.9998999834060669]	الإنكليز

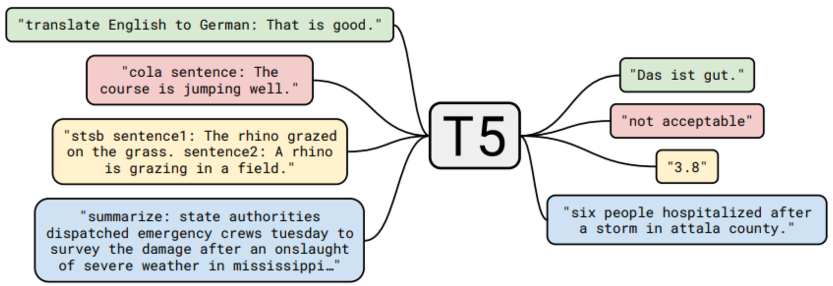


Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

1. Text summarization
2. Question answering
3. Translation
4. Sentiment analysis
5. Natural Language inference
6. Coreference resolution
7. Sentence Completion
8. Word sense disambiguation

Every T5 Task with explanation:

Task Name	Explanation
1.CoLA	Classify if a sentence is grammatically correct
2.RTE	Classify whether a statement can be deducted from a sentence
3.MNLI	Classify for a hypothesis and premise whether they contradict or contradict each other or neither of both (3 class).
4.MRPC	Classify whether a pair of sentences is a re-phrasing of each other (semantically equivalent)
5.QNLI	Classify whether the answer to a question can be deducted from an answer candidate.
6.QQP	Classify whether a pair of questions is a re-phrasing of each other (semantically equivalent)
7.SST2	Classify the sentiment of a sentence as positive or negative
8.STSB	Classify the sentiment of a sentence on a scale from 1 to 5 (21 Sentiment classes)
9.CB	Classify for a premise and a hypothesis whether they contradict each other or not (binary).
10.COPA	Classify for a question, premise, and 2 choices which choice the correct choice is (binary).
11.MultiRc	Classify for a question, a paragraph of text, and an answer candidate, if the answer is correct (binary),
12.WiC	Classify for a pair of sentences and a disambigous word if the word has the same meaning in both sentences.
13.WSC/DPR	Predict for an ambiguous pronoun in a sentence what it is referring to.
14.Summarization	Summarize text into a shorter representation.
15.SQuAD	Answer a question for a given context.
16.WMT1.	Translate English to German
17.WMT2.	Translate English to French
18.WMT3.	Translate English to Romanian

GLUE													CNN/DM			SQuAD		SuperGLUE												WMT		
Score Average	CoLA MCC	SST-2 Acc	MRPC F1	MRPC Acc	STSB PCC	STSB SCC	QQP F1	QQP Acc	$MNLI_m$ Acc	$MNLI_{mm}$ Acc	QNLI Acc	RTE Acc	R-1-F	R-2-F	R-L-F	EM	F1	Score Average	BoolQ Acc	CB F1	CB Acc	COPA Acc	MultiRC F1	MultiRC EM	ReCoRD F1	ReCoRD EM	RTE Acc	WiC Acc	WSC Acc	EnDe BLEU	EnFr BLEU	EnRo BLEU

		GLUE													CNN/DM			SQuAD		SuperGLUE												WMT		
Table	Experiment	Score Average	CoLA MCC	SST-2 Acc	MRPC F1	MRPC Acc	STSB PCC	STSB SCC	QQP F1	QQP Acc	MNLI$_m$ Acc	MNLI$_{mm}$ Acc	QNLI Acc	RTE Acc	R-1-F	R-2-F	R-L-F	EM	F1	Score Average	BoolQ Acc	CB F1	CB Acc	COPA Acc	MultiRC F1	MultiRC EM	ReCoRD F1	ReCoRD EM	RTE Acc	WiC Acc	WSC Acc	EnDe BLEU	EnFr BLEU	EnRo BLEU
1	★ Baseline average	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
1	Baseline standard deviation	0.235	1.111	0.569	0.729	1.019	0.374	0.418	0.108	0.070	0.291	0.231	0.361	1.393	0.065	0.065	0.058	0.343	0.226	0.416	0.365	3.237	2.560	2.741	0.716	1.011	0.370	0.379	1.228	0.850	2.029	0.112	0.090	0.108
1	No pre-training	66.22	12.29	80.62	81.42	73.04	72.58	72.97	81.94	86.62	68.02	67.98	75.69	58.84	39.19	17.60	36.69	50.31	61.97	53.04	65.38	71.61	76.79	62.00	59.10	0.84	20.33	17.95	54.15	54.08	65.38	25.86	39.77	24.04
2	★ Enc/dec, denoising	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
2	Enc/dec, shared, denoising	82.81	55.24	91.86	91.58	88.24	87.43	87.58	88.69	91.60	83.88	84.01	90.23	73.65	41.11	18.78	38.48	80.63	88.49	70.73	77.13	95.04	96.43	65.00	66.16	22.98	68.95	68.09	70.76	68.18	75.96	26.72	39.03	27.46
2	Enc/dec, 6 layers, denoising	80.88	46.26	92.09	91.51	87.99	87.01	86.76	87.93	90.97	82.20	82.41	88.83	71.48	40.83	18.97	38.31	77.59	86.07	68.42	73.79	91.70	92.86	67.00	61.02	19.62	61.26	60.33	72.20	65.99	75.00	26.38	38.40	26.95
2	Language model, denoising	74.70	24.50	90.60	86.08	78.92	85.22	85.42	85.40	88.99	76.72	77.05	86.02	64.62	39.49	17.93	36.91	61.14	71.37	55.02	65.47	60.08	71.43	58.00	43.03	2.94	53.35	52.31	53.07	58.62	63.46	25.09	35.28	25.86
2	Prefix LM, denoising	81.82	49.99	92.43	91.43	88.24	87.20	86.98	88.41	91.39	82.32	82.93	88.71	74.01	40.46	18.61	37.90	78.94	87.31	68.11	75.50	93.37	91.07	60.00	63.43	21.20	65.03	64.11	71.48	65.67	73.08	26.43	37.98	27.39
2	Enc/dec, LM	79.56	42.03	91.86	91.64	88.24	87.13	87.00	88.21	91.15	81.68	81.66	88.54	65.70	40.67	18.59	38.13	76.02	84.85	64.29	72.23	85.74	89.29	57.00	60.53	16.26	59.28	58.30	65.34	64.89	70.19	26.27	39.17	26.86
2	Enc/dec, shared, LM	79.60	44.83	92.09	90.20	85.78	86.03	85.87	87.77	91.02	81.74	82.29	89.16	65.34	40.16	18.13	37.59	76.35	84.86	63.50	70.49	91.41	87.50	55.00	60.21	16.89	57.83	56.73	63.54	63.48	70.19	26.62	39.17	27.05
2	Enc/dec, 6 layers, LM	78.67	38.72	91.40	90.40	86.52	86.82	86.49	87.87	91.03	80.99	80.92	88.05	65.70	40.29	18.26	37.70	75.32	84.06	64.06	71.38	85.25	89.29	60.00	57.56	16.79	55.22	54.30	66.79	63.95	71.15	26.13	38.42	26.89
2	Language model, LM	73.78	28.53	89.79	85.23	78.68	84.22	84.00	84.88	88.70	74.94	75.77	84.84	58.84	38.97	17.54	36.37	53.81	64.55	56.51	64.22	59.92	71.43	64.00	53.04	1.05	46.81	45.78	58.84	56.74	69.23	25.23	34.31	25.38
2	Prefix LM, LM	79.68	41.26	92.09	90.11	86.27	86.82	86.32	88.35	91.35	81.71	82.02	89.04	68.59	39.66	17.84	37.13	76.87	85.39	64.86	71.47	93.37	91.07	57.00	58.67	16.89	59.25	58.16	64.26	66.30	71.15	26.28	37.51	26.76
4	Language modeling with prefix	80.69	44.22	93.00	91.68	88.48	87.20	87.18	88.39	91.41	82.66	83.09	89.29	68.95	40.71	18.94	38.15	77.99	86.43	65.27	73.55	83.95	87.50	55.00	59.65	18.89	61.76	60.76	68.59	65.67	73.08	26.86	39.73	27.49
4	BERT-style (Devlin et al., 2018)	82.96	52.49	92.55	92.79	89.95	87.68	87.66	88.47	91.44	83.60	84.05	90.33	75.45	41.27	19.17	38.72	80.65	88.24	69.85	76.48	94.37	94.64	61.00	63.29	25.08	66.76	65.85	72.20	69.12	75.00	26.78	40.03	27.41
4	Deshuffling	73.17	22.82	87.16	86.88	81.13	84.03	83.82	86.38	89.90	76.30	76.34	84.18	58.84	40.75	18.59	38.10	67.61	76.76	58.47	69.17	63.70	78.57	56.00	59.85	12.70	45.52	44.36	57.04	64.89	68.27	26.11	39.30	25.62
5	BERT-style (Devlin et al., 2018)	82.96	52.49	92.55	92.79	89.95	87.68	87.66	88.47	91.44	83.60	84.05	90.33	75.45	41.27	19.17	38.72	80.65	88.24	69.85	76.48	94.37	94.64	61.00	63.29	25.08	66.76	65.85	72.20	69.12	75.00	26.78	40.03	27.41
5	MASS-style (Song et al., 2019)	82.32	47.01	91.63	92.53	89.71	88.21	88.18	88.58	91.44	82.96	83.67	90.02	77.26	41.16	19.16	38.55	80.10	88.07	69.28	75.08	84.98	89.29	63.00	64.46	23.50	66.71	65.91	72.20	67.71	78.85	26.79	39.89	27.55
5	★ Replace corrupted spans	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
5	Drop corrupted tokens	84.44	60.04	92.89	92.79	89.95	87.28	86.85	88.56	91.54	83.94	83.92	90.74	79.42	41.27	19.31	38.70	80.52	88.28	68.67	75.90	96.02	94.64	56.00	65.06	23.92	65.54	64.60	71.12	67.40	74.04	27.07	39.76	27.82
6	Corruption rate = 10%	82.82	52.71	92.09	91.55	88.24	88.19	88.15	88.47	91.40	83.50	84.51	90.33	75.45	41.05	19.00	38.53	80.38	88.36	69.55	74.98	92.37	92.86	62.00	66.04	24.66	67.93	67.09	70.76	67.24	75.96	26.87	39.28	27.44
6	★ Corruption rate = 15%	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
6	Corruption rate = 25%	83.00	53.47	93.00	92.44	89.46	87.36	87.36	88.68	91.53	84.44	84.15	90.77	74.01	41.69	19.54	39.14	80.96	88.61	70.48	76.39	93.02	92.86	68.00	65.46	24.66	68.20	67.39	73.65	67.87	72.12	27.04	39.83	27.47
6	Corruption rate = 50%	81.27	46.26	91.63	91.11	87.99	87.87	87.64	88.70	91.57	83.64	84.10	90.24	70.76	41.51	19.32	38.89	79.80	87.76	70.33	75.02	93.05	92.86	68.00	62.97	24.13	64.94	64.13	72.20	68.50	77.88	27.01	39.90	27.49
7	★ Baseline (i.i.d.)	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
7	Average span length = 2	83.54	53.82	92.20	93.05	90.44	87.85	87.71	88.42	91.40	84.28	84.46	90.88	77.62	41.23	19.39	38.69	82.09	89.69	72.20	77.06	90.43	91.07	70.00	66.28	26.13	71.34	70.61	75.45	68.34	78.85	26.76	39.99	27.63
7	Average span length = 3	83.49	53.90	92.43	92.25	89.46	87.49	87.53	88.72	91.51	84.85	84.84	90.99	77.26	41.50	19.62	38.94	81.84	89.66	72.53	76.85	94.37	94.64	70.00	67.64	28.75	70.84	69.90	74.73	67.71	77.88	26.86	39.65	27.62
7	Average span length = 5	83.40	52.12	93.12	92.63	89.71	88.70	88.47	88.84	91.64	84.32	84.29	90.79	76.90	41.39	19.24	38.82	82.05	89.79	72.23	77.06	83.06	89.29	69.00	68.16	30.12	71.36	70.53	75.81	69.91	79.81	26.88	39.40	27.53
7	Average span length = 10	82.85	50.11	92.09	91.95	88.97	88.45	88.22	88.86	91.63	84.34	84.28	91.07	76.17	41.38	19.33	38.80	81.84	89.39	70.44	76.45	87.40	89.29	65.00	66.87	29.59	69.82	68.94	72.56	67.55	75.96	26.79	39.49	27.69
8	★ C4	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
8	C4, unfiltered	81.46	48.01	91.63	92.72	89.95	87.79	87.60	88.31	91.27	82.30	82.34	88.71	72.20	41.09	19.14	38.54	78.78	87.04	68.04	75.75	89.17	91.07	62.00	65.52	25.60	62.42	61.58	69.68	67.08	72.12	26.55	39.34	27.21
8	RealNews-like	83.83	56.55	92.66	92.06	88.97	87.71	87.37	88.51	91.49	84.35	84.46	90.61	78.34	41.38	19.23	38.84	80.39	88.50	72.38	77.00	93.09	94.64	66.00	65.92	23.82	74.56	73.72	75.81	66.61	80.77	26.75	39.90	27.48
8	WebText-like	84.03	56.38	93.12	92.31	89.22	88.69	88.68	88.65	91.56	84.70	84.84	90.83	77.62	41.23	19.31	38.70	81.42	89.15	71.40	76.88	83.08	89.29	66.00	64.10	24.24	72.24	71.36	75.45	68.03	82.69	26.80	39.74	27.59
8	Wikipedia	81.85	45.53	92.32	91.67	88.24	85.62	86.40	88.37	91.34	82.61	83.25	90.96	77.26	41.39	19.31	38.81	81.29	89.18	68.01	76.12	56.03	80.36	67.00	65.01	25.92	69.03	68.06	74.73	67.08	76.92	26.94	39.69	27.67
8	Wikipedia + TBC	83.65	55.53	92.78	92.41	89.22	86.67	86.27	89.47	92.29	84.38	83.45	91.94	76.90	41.22	19.28	38.67	82.08	89.70	73.24	76.22	95.40	92.86	69.00	51.59	50.93	69.53	68.51	77.62	66.93	81.73	26.77	39.63	27.57
9	★ Full data set	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
9	2^{29} (64 repeats)	82.87	53.82	92.78	91.79	88.73	87.56	87.58	88.73	91.54	84.07	84.21	90.59	73.65	41.18	19.19	38.67	80.97	88.90	72.03	76.76	92.96	92.86	66.00	65.11	26.76	69.35	68.49	75.81	67.24	82.69	26.83	39.74	27.63
9	2^{27} (256 repeats)	82.62	50.60	92.32	92.07	88.73	87.83	87.60	88.65	91.54	83.43	84.37	90.12	75.81	41.24	19.20	38.70	79.78	87.63	69.97	75.29	93.42	91.07	63.00	61.82	23.61	66.27	65.39	73.65	66.30	80.77	27.02	39.71	27.33
9	2^{25} (1,024 repeats)	79.55	43.84	91.28	89.32	85.05	85.92	85.74	88.05	91.09	81.29	81.72	87.90	69.31	40.66	18.57	38.13	76.27	84.58	64.76	72.63	83.97	82.14	64.00	59.39	17.94	56.94	56.04	64.98	65.20	73.08	26.38	39.56	26.80
9	2^{23} (4,096 repeats)	76.34	32.68	89.45	89.84	86.03	83.49	83.42	87.18	90.61	77.80	78.69	85.47	64.62	40.16	18.33	37.66	70.92	80.20	59.29	69.85	73.48	73.21	56.00	57.66	14.38	46.69	45.79	59.57	65.05	68.27	26.37	38.84	25.81
10	★ All parameters	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
10	Adapter layers, $d = 32$	80.52	45.33	91.63	90.59	86.76	88.38	88.06	86.99	90.26	83.63	83.94	90.72	67.15	34.50	15.08	32.15	79.32	87.70	60.40	65.32	50.87	73.21	52.00	58.61	19.41	65.50	64.58	62.09	64.58	73.08	13.84	17.88	15.54
10	Adapter layers, $d = 128$	81.51	45.35	92.89	91.49	88.24	87.73	87.65	87.73	90.93	83.64	84.09	90.52	72.56	36.71	16.62	34.37	79.47	87.61	63.03	69.20	52.21	75.00	56.00	61.08	18.05	67.94	66.97	68.59	66.77	73.08	19.83	27.50	22.63
10	Adapter layers, $d = 512$	81.54	44.25	93.35	91.00	87.25	88.74	88.44	88.02	91.15	83.08	83.80	89.62	74.37	38.63	17.78	36.25	79.18	87.32	64.30	73.18	59.86	71.43	56.00	62.94	18.57	66.56	65.74	70.76	67.87	74.04	23.45	33.98	25.81
10	Adapter layers, $d = 2048$	82.62	49.86	92.55	91.30	87.99	88.46	88.35	88.36	91.40	83.63	83.18	90.66	76.53	39.44	18.30	37.06	79.40	87.36	68.61	74.53	88.00	91.07	58.00	61.10	18.89	66.73	66.06	73.29	71.16	75.96	25.64	36.92	26.93
10	Gradual Unfreezing	82.50	51.74	91.97	92.61	89.71	87.27	86.90	88.26	91.35	83.42	83.49	89.71	75.09	40.88	18.95	38.40	79.17	87.30	70.79	75.51	93.09	94.64	70.00	62.03	21.51	65.69	64.79	72.92	69.12	77.89	26.71	39.02	26.93
11	★ Baseline (pre-train/fine-tune)	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
11	Equal	76.13	39.47	90.94	82.90	75.74	78.83	78.44	86.45	89.71	82.08	82.92	90.13	59.93	40.95	19.02	38.39	76.51	85.61	63.37	73.06	82.37	83.93	65.00	60.89	17.52	60.51	59.70	61.01	60.03	65.38	23.89	34.31	26.78
11	Examples-proportional, $K = 2^{16}$	80.45	42.07	91.97	90.97	87.50	85.41	85.04	86.89	90.10	83.01	83.66	90.74	72.56	41.16	19.04	38.59	77.25	85.72	69.95	76.67	86.38	89.29	70.00	65.93	27.91	62.78	61.95	76.90	65.83	73.08	24.35	34.99	27.10
11	Examples-proportional, $K = 2^{17}$	81.56	47.35	91.40	91.55	88.24	86.15	85.93	86.94	90.06	82.76	84.12	90.79	75.09	41.06	19.12	38.47	77.00	85.87	67.91	77.89	77.54	85.71	57.00	67.78	27.07	61.51	60.54	79.06	65.20	74.04	24.36	35.00	27.25
11	Examples-proportional, $K = 2^{18}$	81.67	46.85	91.63	91.99	88.73	87.68	87.20	86.93	90.35	83.30	84.01	91.47	73.29	40.96	19.07	38.43	78.17	86.74	67.94	76.57	78.88	87.50	62.00	67.70	30.85	63.43	62.54	76.53	65.67	67.31	24.57	35.19	27.39
11	Examples-proportional, $K = 2^{19}$	81.42	45.94	91.63	92.20	89.22	88.44	88.32	86.84	90.10	83.73	84.29	91.84	70.40	41.26	19.24	38.71	79.78	88.15	67.30	75.66	75.59	87.50	59.00	68.22	30.64	65.32	64.29	73.65	65.05	69.23	25.21	36.30	27.76
11	Examples-proportional, $K = 2^{20}$	80.80	42.55	92.78	91.27	87.99	88.36	88.10	86.10	89.62	84.15	84.26	92.20	68.95	41.05	19.24	38.46	80.36	88.27	67.38	73.21	76.18	83.93	62.00	67.57	26.86	66.12	65.22	76.90	64.73	69.23	25.66	36.93	27.68
11	Examples-proportional, $K = 2^{21}$	79.83	44.45	91.28	89.00	84.31	87.54	87.40	84.93	88.53	82.54	84.16	90.85	67.87	40.51	18.79	37.92	79.50	87.48	65.10	71.16	68.88	85.71	57.00	62.75	23.40	64.50	63.65	72.92	64.11	71.15	25.82	37.22	27.13
11	Temperature-scaled, $T = 2$	81.90	54.00	91.74	90.56	86.76	85.11	84.60	86.40	89.74	83.47	84.15	91.51	72.56	41.09	19.28	38.54	79.42	87.77	69.92	76.73	92.37	92.86	57.00	69.80	31.90	66.65	65.74	72.92	67.08	75.96	25.42	36.72	27.20
11	Temperature-scaled, $T = 4$	80.56	45.38	91.97	89.68	85.78	83.13	82.76	86.39	90.00	82.78	84.19	91.16	73.65	41.09	19.22	38.51	77.99	86.81	69.54	76.76	97.36	96.43	59.00	68.10	31.48	64.26	63.27	74.73	64.26	71.15	25.04	35.82	27.45
11	Temperature-scaled, $T = 8$	77.21	40.07	91.06	88.11	83.33	79.20	79.06	86.60	89.90	83.05	83.56	90.21	59.93	41.01	19.10	38.40	77.14	85.99	66.07	73.94	93.70	94.64	60.00	66.36	26.86	63.46	62.60	62.09	63.32	65.38	24.55	35.35	27.17
12	★ Unsupervised pre-training + fine-tuning	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
12	Multi-task training	81.42	45.94	91.63	92.20	89.22	88.44	88.32	86.84	90.10	83.73	84.29	91.84	70.40	41.26	19.24	38.71	79.78	88.15	67.30	75.66	75.59	87.50	59.00	68.22	30.64	65.32	64.29	73.65	65.05	69.23	25.21	36.30	27.76
12	Multi-task pre-training + fine-tuning	83.11	51.42	92.66	91.73	88.73	88.06	87.70	88.61	91.61	84.09	84.31	91.85	76.53	41.15	19.12	38.59	80.26	88.50	71.03	79.54	81.69	87.50	65.00	70.72	31.48	65.94	65.03	81.23	68.18	73.08	27.08	39.80	28.07
12	Leave-one-out multi-task training	81.98	48.00	93.23	91.72	88.24	87.76	87.32	88.61	91.44	84.00	84.11	90.79	72.20	41.34	19.05	38.77	79.97	88.10	71.68	78.35	86.76	89.29	66.00	68.09	29.49	66.23	65.27	79.06	68.65	78.85	26.93	39.79	27.87
12	Supervised multi-task pre-training	79.93	36.60	92.43	91.58	88.24	87.03	86.78	88.15	91.20	82.87	83.16	90.13	70.76	41.12	18.96	38.49	77.38	85.65	65.36	75.66	68.87	83.93	58.00	64.81	21.93	55.37	54.61	71.12	67.40	75.96	26.81	40.13	28.04
13	★ Baseline	83.28	53.84	92.68	92.07	88.92	88.02	87.94	88.67	91.56	84.24	84.57	90.48	76.28	41.33	19.24	38.77	80.88	88.81	71.36	76.62	91.22	91.96	66.20	66.13	25.78	69.05	68.16	75.34	68.04	78.56	26.98	39.82	27.65
13	1× size, 4× training steps	85.33	60.29	93.81	94.06	91.67	89.42	89.25	89.15	91.87	86.01	85.70	91.63	78.34	41.52	19.33	38.96	82.45	90.19	74.72	79.17	94.75	92.86	71.00	67.34	29.70	72.63	71.59	78.34	72.10	82.69	27.08	40.66	27.93
13	1× size, 4× batch size	84.60	56.08	93.12	92.31	89.22	88.85	88.84	89.35	92.07	85.98	86.13	91.07	80.14	41.70	19.42	39.08	82.52	90.21	74.64	78.78	93.69	94.64	72.00	68.09	30.95	74.73	73.90	76.53	70.06	81.73	27.07	40.60	27.84
13	2× size, 2× training steps	86.18	62.04	93.69	93.36	90.69	89.18	89.23	89.35	92.05	87.23	87.05	92.68	81.95	41.74	19.66	39.14	84.18	91.29	77.18	80.98	97.36	96.43	74.00	71.34	35.68	77.11	76.34	80.51	69.28	85.58	27.52	41.03	28.19
13	4× size, 1× training steps	85.91	57.58	94.38	92.67	89.95	89.60	89.60	89.44	92.14	87.05	87.12	93.12	83.39	41.60	19.73	39.08	83.86	91.32	78.04	81.38	89.09	94.64	73.00	73.74	40.40	78.25	77.40	81.59	70.22	91.35	27.47	40.71	28.10
13	4× ensembled	84.77	56.14	93.46	93.31	90.67	89.71	89.60	89.62	92.24	86.22	86.53	91.60	77.98	42.10	20.10	39.56	83.09	90.40	71.74	77.58	89.85	91.07	66.00	69.32	29.49	72.67	71.94	76.90	69.12	72.12	28.05	40.53	28.09
13	4× ensembled, fine-tune only	84.05	54.78	92.78	93.15	90.44	88.34	88.12	89.27	91.97	85.33	85.88	90.98	77.62	41.66	19.57	39.12	82.36	89.86	71.56	77.43	90.07	92.86	69.00	67.31	26.34	70.47	69.64	75.45	68.18	74.04	27.55	40.22	28.09

109 Languages supported by **Language-agnostic BERT Sentence Embedding** (LABSE)

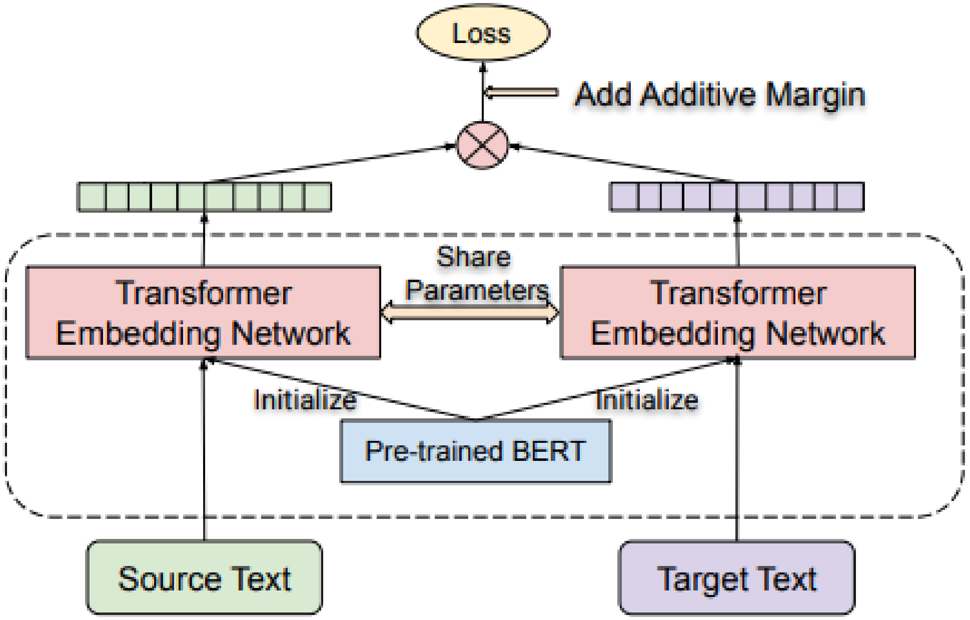


Figure 1: Dual encoder model with BERT based encoding modules.

ISO	NAME	ISO	NAME	ISO	NAME
af	AFRIKAANS	ht	HAITIAN_CREOLE	pt	PORTUGUESE
am	AMHARIC	hu	HUNGARIAN	ro	ROMANIAN
ar	ARABIC	hy	ARMENIAN	ru	RUSSIAN
as	ASSAMESE	id	INDONESIAN	rw	KINYARWANDA
az	AZERBAIJANI	ig	IGBO	si	SINHALESE
be	BELARUSIAN	is	ICELANDIC	sk	SLOVAK
bg	BULGARIAN	it	ITALIAN	sl	SLOVENIAN
bn	BENGALI	ja	Japanese	sm	SAMOAN
bo	TIBETAN	jv	JAVANESE	sn	SHONA
bs	BOSNIAN	ka	GEORGIAN	so	SOMALI
ca	CATALAN	kk	KAZAKH	sq	ALBANIAN
ceb	CEBUANO	km	KHMER	sr	SERBIAN
co	CORSICAN	kn	KANNADA	st	SESOTHO
cs	CZECH	ko	KOREAN	su	SUNDANESE
cy	WELSH	ku	KURDISH	sv	SWEDISH
da	DANISH	ky	KYRGYZ	sw	SWAHILI
de	GERMAN	la	LATIN	ta	TAMIL
el	GREEK	lb	LUXEMBOURGISH	te	TELUGU
en	ENGLISH	lo	LAOTHIAN	tg	TAJIK
eo	ESPERANTO	lt	LITHUANIAN	th	THAI
es	SPANISH	lv	LATVIAN	tk	TURKMEN
et	ESTONIAN	mg	MALAGASY	tl	TAGALOG
eu	BASQUE	mi	MAORI	tr	TURKISH
fa	PERSIAN	mk	MACEDONIAN	tt	TATAR
fi	FINNISH	ml	MALAYALAM	ug	UIGHUR
fr	FRENCH	mn	MONGOLIAN	uk	UKRAINIAN
fy	FRISIAN	mr	MARATHI	ur	URDU
ga	IRISH	ms	MALAY	uz	UZBEK
gd	SCOTS_GAELIC	mt	MALTESE	vi	VIETNAMESE
gl	GALICIAN	my	BURMESE	wo	WOLOF
gu	GUJARATI	ne	NEPALI	xh	XHOSA
ha	HAUSA	nl	DUTCH	yi	YIDDISH
haw	HAWAIIAN	no	NORWEGIAN	yo	YORUBA
he	HEBREW	ny	NYANJA	zh	Chinese
hi	HINDI	or	ORIYA	zu	ZULU
hmn	HMONG	pa	PUNJABI		
hr	CROATIAN	pl	POLISH		

MARIANNMT

Fast Neural Machine Translation in C++

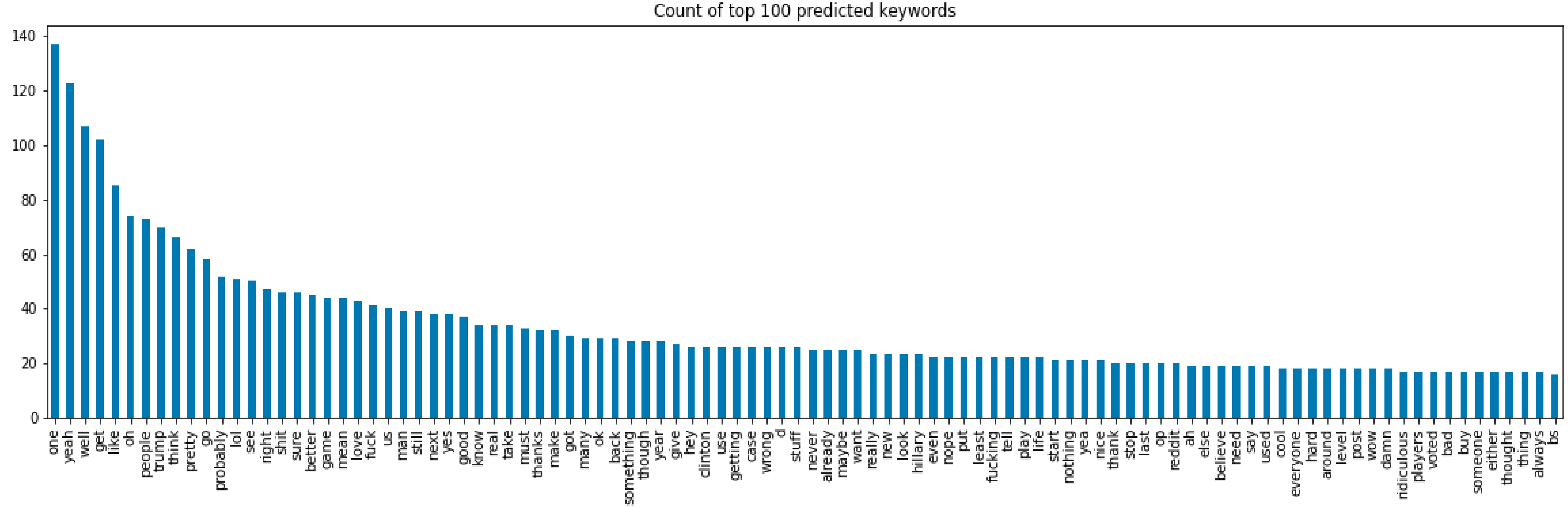
Translate between 200+ Languages
With Marian: Fast Neural Machine Translation in C++

Microsoft

Extract keywords and plot distribution with YAKE

```
keyword_predictions.explode('keywords').keywords.value_counts()[0:100].plot.bar(title='Count of top 100 predicted keywords', figsize=(20,5))
```

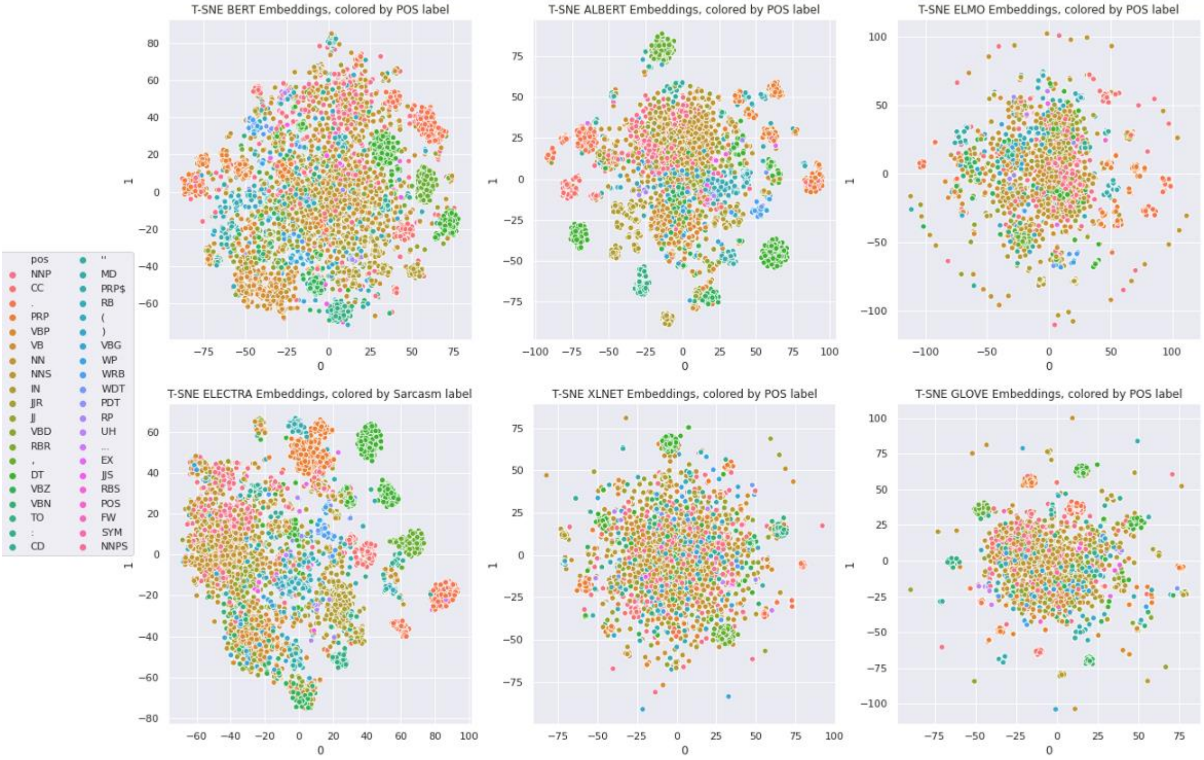
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8a3021c550>
```



t-SNE Visualizations with NLU

- t-SNE [1] is a tool to visualize high-dimensional data. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data.
- Pefect to visualize high dimensional wordembeddings generated from a dataset

Made with 1 NLU line and a few lines of plotting:



[1] van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data

Demo time

What does NLU 1.1 include?

100% Open Source	1000+ pre-trained models	**100+** of the latest NLP word embeddings (BERT, ELMO, ALBERT, XLNET, GLOVE, BIOBERT, ELECTRA, COVIDBERT) and different variations of them	**50+** of the latest NLP sentence embeddings (BERT, ELECTRA, USE) and different variations of them	Multilingal Sentence and Word embeddings
50+ Classifiers	200 + Supported Languages	Labeled and Unlabeled Dependency parsing	Spell Checking	All in one line of code !
Unsupervised Keyword Extraction with YAKE	Various text-preprocessing and cleaning methods	Summarize	Answer questions	SQUAD/GLUE/SUPERGLUE
Train Classifiers	State of the art	Aspect NER	Aspect Sentiment	Intent Classification

Afrikaans (af)
Arabic (ar)
Azeri (az)
Bulgarian (bg)
Bislama (bi)
Bengali (bn)
Breton (br)
Catalan (ca)
Czech (cs)
Welsh (cy)
Danish (da)
German (de)
Ewe (ee)
Greek (el)
English (en)
Esperanto (eo)
Spanish (es)
Estonian (et)
Basque (eu)
Farsi (fa)
Finnish (fi)
Fiji (fj)
French (fr)
Irish (ga)
Galician (gl)
Manx (gv)
Hausa (ha)
Hebrew (he)
Hindi (hi)
Hiri Motu (ho)
Haitian (ht)
Hungarian (hu)
Armenian (hy)
Indonesian (id)
Igbo (ig)
Icelandic (is)
Italian (it)
Japanese (ja)
Georgian (ka)
Kongo (kg)
Kuanyama (kj)
Greenlandic (kl)
Korean (ko)
Latin (la)
Ganda (lg)
Lingala (ln)
Luba-Katanga (lu)
Latvian (lv)
Malagasy (mg)
Marshallese (mh)
FYRO Macedonian (mk)
Malayalam (ml)
Marathi (mr)
Maltese (mt)
Ndonga (ng)
Dutch (nl)
Norwegian (Bokmal) (no)
Chichewa (ny)
Oromoor (om)
Punjabi (pa)
Polish (pl)
Portuguese (pt)
Kirundi (rn)
Romanian (ro)
Russian (ru)
Kinyarwanda (rw)
Sangro (sg)
Slovak (sk)
Slovenian (sl)
Samoan (sm)
Shona (sn)
Somali (so)
Albanian (sq)
Siswati (ss)
Sesotho (st)
Swedish (sv)
Thai (th)
Tigrinya (ti)
Tagalog (tl)
Tswana (tn)
Tonga (to)
Turkish (tr)
Tsonga (ts)
Twi (tw)
Tahitian (ty)
Ukrainian (uk)
Urdu (ur)
Venda (ve)
Vietnamese (vi)
Walloon (wa)
Xhosa (xh)
Yoruba (yo)
Chinese (zh)
Zulu (zu)

1000+ MODELS & PIPELINES, 300+ LANGUAGES, 2-WEEK RELEASES

Annotators

Annotator
Tokenizer*
Normalizer*
Stemmer*
Lemmatizer*
RegexMatcher*
TextMatcher*
Chunker*
DateMatcher*
SentenceDetector*
DeepSentenceDetector*
POSTagger
ViveknSentimentDetector
SentimentDetector*
WordEmbeddings*
BertEmbeddings*
NerCrf
NerDL
NorvigSweeting
SymmetricDelete
ContextSpellChecker
DependencyParser
TypedDependencyParser
AssertionLogReg
AssertionDL
EntityResolver
DeIdentification

Pretrained Models

Model	Name	Version	Language
LemmatizerModel (Lemmatizer)	lemma_antbnc	Open Source	English
PerceptronModel (POS)	pos_anc	Open Source	English
NerCRFModel (NER with GloVe)	ner_crf	Open Source	English
NerDLModel (NER with GloVe)	ner_dl	Open Source	English
NerDLModel (NER with GloVe)	ner_dl_contrib	Open Source	English
NerDLModel (NER with BERT)	ner_dl_bert_base_cased	Open Source	English
NerDLModel (OntoNotes with GloVe 100d)	onto_100	Open Source	English
NerDLModel (OntoNotes with GloVe 300d)	onto_300	Open Source	English
WordEmbeddings (GloVe)	glove_100d	Open Source	English
BertEmbeddings (base_uncased)	bert_base_uncased	Open Source	English
BertEmbeddings (base_cased)	bert_base_cased	Open Source	English
BertEmbeddings (large_uncased)	bert_large_uncased	Open Source	English
BertEmbeddings (large_cased)	bert_large_cased	Open Source	English
DeepSentenceDetector	ner_dl_sentence	Open Source	English
ContextSpellCheckerModel (Spell Checker)	spellcheck_dl	Open Source	English
SymmetricDeleteModel (Spell Checker)	spellcheck_sd	Open Source	English
NorvigSweetingModel (Spell Checker)	spellcheck_norvig	Open Source	English
ViveknSentimentModel (Sentiment)	sentiment_vivekn	Open Source	English
DependencyParser (Dependency)	dependency_conllu	Open Source	English
TypedDependencyParser (Dependency)	dependency_typed_conllu	Open Source	English
NerDLModel	ner_clinical	Licensed	English
AssertionLogRegModel	assertion_ml	Licensed	English
AssertionDLModel	assertion_dl	Licensed	English
NerDLModel	deidentify_dl	Licensed	English
DeIdentificationModel	deidentify_rb	Licensed	English
WordEmbeddingsModel	embeddings_clinical	Licensed	English
PerceptronModel	pos_clinical	Licensed	English
EntityResolverModel	resolve_icd10	Licensed	English
EntityResolverModel	resolve_icd10cm_cl_em	Licensed	English
EntityResolverModel	resolve_icd10pcs_cl_em	Licensed	English
ContextSpellCheckerModel	context_spell_med	Licensed	English
LemmatizerModel (Lemmatizer)	lemma	Open Source	French
PerceptronModel (POS UD)	pos_ud_gsd	Open Source	French
NerDLModel (glove_840B_300)	wikiner_840B_300	Open Source	French
LemmatizerModel (Lemmatizer)	lemma	Open Source	German
PerceptronModel (POS UD)	pos_ud_hdt	Open Source	German
NerDLModel (glove_840B_300)	wikiner_840B_300	Open Source	German
LemmatizerModel (Lemmatizer)	lemma_dxc	Open Source	Italian
SentimentDetector (Sentiment)	sentiment_dxc	Open Source	Italian
PerceptronModel (POS UD)	pos_ud_isdt	Open Source	Italian
NerDLModel (glove_840B_300)	wikiner_840B_300	Open Source	Italian
WordEmbeddings (GloVe)	glove_840B_300	Open Source	Multi-language
WordEmbeddings (GloVe)	glove_6B_300	Open Source	Multi-language
BertEmbeddings (multi_cased)	bert_multi_cased	Open Source	Multi-language

Pretrained pipelines

Pipelines	Name	Language
Explain Document ML	explain_document_ml	English
Explain Document DL	explain_document_dl	English
Explain Document DL Win	explain_document_dl_noncontrib	English
Explain Document DL Fast	explain_document_dl_fast	English
Explain Document DL Fast Win	explain_document_dl_fast_noncontrib	English
Recognize Entities DL	recognize_entities_dl	English
Recognize Entities DL Win	recognize_entities_dl_noncontrib	English
OntoNotes Entities Small	onto_recognize_entities_sm	English
OntoNotes Entities Large	onto_recognize_entities_lg	English
Match Datetime	match_datetime	English
Match Pattern	match_pattern	English
Match Chunk	match_chunks	English
Match Phrases	match_phrases	English
Clean Stop	clean_stop	English
Clean Pattern	clean_pattern	English
Clean Slang	clean_slang	English
Check Spelling	check_spelling	English
Analyze Sentiment	analyze_sentiment	English
Dependency Parse	dependency_parse	English
Explain Document Large	explain_document_lg	French
Explain Document Medium	explain_document_md	French
Entity Recognizer Large	entity_recognizer_lg	French
Entity Recognizer Medium	entity_recognizer_md	French
Explain Document Large	explain_document_lg	Italian
Explain Document Medium	explain_document_md	Italian
Entity Recognizer Large	entity_recognizer_lg	Italian
Entity Recognizer Medium	entity_recognizer_md	Italian

Thank you.

christian@JohnSnowLabs.com

@ckl_it

In/Christian-Kasim-Loan

Medium.com/@Christian.Kasim.Loan