Trankit
A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen

EACL 2021: System Demonstrations
Multilingual Natural Language Processing

- A real need to build Multilingual NLP systems to overcome language barriers (Aharoni et al., 2019; Liu et al., 2019a; Taghizadeh and Faili, 2020; Zhu, 2020; Kanayama and Iwamoto, 2020; Nguyen and Nguyen, 2021).
- Such systems usually rely on upstream features: sentence and token boundaries, parts of speech, dependency trees, ...

English:
I would put it out there anyways.

Chinese:
然而，这样的处理也衍生了一些问题。

Arabic:
سورية: تعداد وزارو واسع يشمل 8 حقائق

Vietnamese:
Việt Nam đẹp tuyệt vời!

Upstream processing
- Sentence splitting
- Tokenization
- Part-of-speech tagging
- Dependency parsing
- ...

Multilingual NLP Downstream system

Output
Multilingual Natural Language Processing

- A real need to build Multilingual NLP systems to overcome language barriers (Aharoni et al., 2019; Liu et al., 2019a; Taghizadeh and Faili, 2020; Zhu, 2020; Kanayama and Iwamoto, 2020; Nguyen and Nguyen, 2021).
- Such systems usually rely on upstream features: sentence and token boundaries, parts of speech, dependency trees, …

English: I would put it out there anyways.

Chinese: 然而，这样的处理也衍生了一些问题。

Arabic: سوريا: تعداد وزاري واسع يشمل 8 حقائق

Vietnamese: Việt Nam đẹp tuyệt vời!

Upstream processing:
- Sentence splitting
- Tokenization
- Part-of-speech tagging
- Dependency parsing
- …
Limitations

- Popular Multilingual NLP Toolkits: Spacy *(spacy.io)*, Flair *(Akbik et al., 2019)*, Udify *(Kondratyuk and Straka, 2019)*, CoreNLP *(Manning et al., 2014)*, UDPipe *(Straka, 2018)*, Stanza *(Qi et al., 2020)*.

<table>
<thead>
<tr>
<th>Toolkit</th>
<th>Raw text processing?</th>
<th>State-of-the-art performance</th>
<th>Supported languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spacy</td>
<td>x</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Flair</td>
<td></td>
<td>x</td>
<td>12</td>
</tr>
<tr>
<td>UDify</td>
<td></td>
<td></td>
<td>75</td>
</tr>
<tr>
<td>CoreNLP</td>
<td>x</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>UDPipe</td>
<td>x</td>
<td>x</td>
<td>61</td>
</tr>
<tr>
<td>Stanza</td>
<td>x</td>
<td>x</td>
<td>66</td>
</tr>
</tbody>
</table>
Limitations

- UDPipe (Straka, 2018) and Stanza (Qi et al., 2020) are currently the state-of-the-art multilingual NLP toolkits:
  - Perform sentence and token splitting at character level.

  “John Donovan from Argghhh! has put out a excellent slide show on what was actually found and fought for in Fallujah.”
Limitations

- UDPipe *(Straka, 2018)* and Stanza *(Qi et al., 2020)* are currently the state-of-the-art multilingual NLP toolkits:
  - Perform sentence and token splitting at character level.
  - Ignore contextualized embeddings for word-level tasks.
  - Train and load separate embeddings (i.e., word2vec, fastText) for different languages.

<table>
<thead>
<tr>
<th>Model Package</th>
<th>Stanza</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>393.9MB</td>
</tr>
<tr>
<td>Chinese</td>
<td>225.2MB</td>
</tr>
<tr>
<td>English</td>
<td>383.5MB</td>
</tr>
<tr>
<td>French</td>
<td>561.9MB</td>
</tr>
<tr>
<td>Spanish</td>
<td>556.1MB</td>
</tr>
<tr>
<td><strong>Total size for 5 languages</strong></td>
<td>~ 2 GB</td>
</tr>
<tr>
<td><strong>Total size for 50 languages</strong></td>
<td>~ 20 GB</td>
</tr>
</tbody>
</table>
Trankit: A Multilingual NLP Toolkit

- First **transformer-based** token and sentence splitter for **56 languages**.
- **90 transformer-based pipelines** trained on Universal Dependencies (UD) v2.5 ([Zeman et al., 2019](#)), significantly outperforming previous SOTA (i.e., Stanza, UDPipe) on different tasks.
- **Solves the memory issue** of transformer-based architecture and the multilingual setting via a novel plug-and-play mechanism with Adapters.
- **Light-weight, highly extensible, and smoothly usable** for general users.
Adapters (Pfeiffer et al., 2020a,b) are small networks injected inside all layers of a pretrained transformer.

During training, the pretrained transformer is fixed while only adapters and task-specific output layers are updated.
System Design

- A set of adapters and task-specific output layers are created for each transformer-based component for each language while only one single large multilingual pretrained transformer is shared across components and languages.
Multilingual Encoder with Adapters

- Given a sequence of subwords \( \mathbf{w} = [w_1, w_2, \ldots, w_K] \), their representations for language \( l \) and component \( m \) are computed by:

\[
x_{1:K}^{l,m} = \text{Transformer}(w_{1:K}; \theta_{AD}^{l,m})
\]

where \( \theta_{AD}^{l,m} \) are the adapters for language \( l \) and component \( m \) of the system.
A set of adapters and task-specific output layers are created for each transformer-based component for each language while only one single large multilingual pretrained transformer is shared across components and languages.
The paint and wheels looked like glass and the interior looked new! Also, they have great
System Design

- A set of adapters and task-specific output layers are created for each transformer-based component for each language while only one single large multilingual pretrained transformer is shared across components and languages.
Joint Model for POS tagging, Morphological Tagging, and Dependency Parsing

The paint and wheels looked like glass and the interior looked new!
Joint Model for POS tagging, Morphological Tagging, and Dependency Parsing

The paint and wheels looked like glass and the interior looked new!

Deep Biaffine Attention

(Dozat and Manning, 2017)

Chu-Liu/Edmonds algorithm

(Chu, 1965; Edmonds, 1967)

Dependent rep

Head rep

Biaffine

Relation score

Multilingual Encoder

Visualization: Brat Rapid Annotation Tool.
A set of adapters and task-specific output layers are created for each transformer-based component for each language while only one single large multilingual pretrained transformer is shared across components and languages.
Named Entity Recognizer

Christopher Nolan is a good film director
Multi-word Token Expander & Lemmatizer

Lemmatization
(Ex: English)

Multi-word Token Expansion
(Ex: French)
System Evaluation

- Performance comparison on Universal Dependencies v2.5 test sets:

<table>
<thead>
<tr>
<th>Treebank</th>
<th>System</th>
<th>Tokens</th>
<th>Sentences</th>
<th>Words</th>
<th>UPOS</th>
<th>XPOS</th>
<th>UFeats</th>
<th>Lemmas</th>
<th>LAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall (90 treebanks)</td>
<td>TranKit</td>
<td>99.23</td>
<td>91.82</td>
<td>99.02</td>
<td>95.65</td>
<td>94.05</td>
<td>93.21</td>
<td>94.27</td>
<td>87.06</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>99.26</td>
<td>88.58</td>
<td>98.90</td>
<td>94.21</td>
<td>92.50</td>
<td>91.75</td>
<td>94.15</td>
<td>83.06</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>99.98</td>
<td>92.09</td>
<td>94.58</td>
<td>90.36</td>
<td>84.00</td>
<td>84.16</td>
<td>88.46</td>
<td>72.67</td>
</tr>
<tr>
<td>Arabic-PADT</td>
<td>TranKit</td>
<td>99.93</td>
<td>96.59</td>
<td>99.22</td>
<td>96.31</td>
<td>94.08</td>
<td>94.28</td>
<td>94.65</td>
<td>88.39</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>99.98</td>
<td>80.43</td>
<td>97.88</td>
<td>94.89</td>
<td>91.75</td>
<td>91.86</td>
<td>93.27</td>
<td>79.33</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>99.98</td>
<td>82.09</td>
<td>94.58</td>
<td>90.36</td>
<td>84.00</td>
<td>84.16</td>
<td>88.46</td>
<td>72.67</td>
</tr>
<tr>
<td>Chinese-GSD</td>
<td>TranKit</td>
<td>97.01</td>
<td>99.7</td>
<td>97.01</td>
<td>94.21</td>
<td>94.02</td>
<td>96.59</td>
<td>97.01</td>
<td>85.19</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>92.83</td>
<td>98.80</td>
<td>92.83</td>
<td>89.12</td>
<td>88.93</td>
<td>92.11</td>
<td>92.83</td>
<td>72.88</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>90.27</td>
<td>99.10</td>
<td>90.27</td>
<td>84.13</td>
<td>84.04</td>
<td>89.05</td>
<td>90.26</td>
<td>61.60</td>
</tr>
<tr>
<td>English-EWT</td>
<td>TranKit</td>
<td>98.48</td>
<td>88.35</td>
<td>98.48</td>
<td>95.95</td>
<td>95.71</td>
<td>96.26</td>
<td>96.84</td>
<td>90.14</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>99.01</td>
<td>81.13</td>
<td>99.01</td>
<td>95.40</td>
<td>95.12</td>
<td>96.11</td>
<td>97.21</td>
<td>86.22</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>98.90</td>
<td>77.40</td>
<td>98.90</td>
<td>93.26</td>
<td>92.75</td>
<td>94.23</td>
<td>95.45</td>
<td>80.22</td>
</tr>
<tr>
<td></td>
<td>spaCy</td>
<td>97.44</td>
<td>63.16</td>
<td>97.44</td>
<td>86.99</td>
<td>91.05</td>
<td>-</td>
<td>87.16</td>
<td>55.38</td>
</tr>
<tr>
<td>French-GSD</td>
<td>TranKit</td>
<td>99.7</td>
<td>96.63</td>
<td>99.66</td>
<td>97.85</td>
<td>-</td>
<td>97.16</td>
<td>97.80</td>
<td>94.00</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>99.68</td>
<td>94.92</td>
<td>99.48</td>
<td>97.30</td>
<td>-</td>
<td>96.72</td>
<td>97.64</td>
<td>91.38</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>99.68</td>
<td>93.59</td>
<td>98.81</td>
<td>95.85</td>
<td>-</td>
<td>95.55</td>
<td>96.61</td>
<td>87.14</td>
</tr>
<tr>
<td></td>
<td>spaCy</td>
<td>99.02</td>
<td>89.73</td>
<td>94.81</td>
<td>89.67</td>
<td>-</td>
<td>-</td>
<td>88.55</td>
<td>75.22</td>
</tr>
<tr>
<td>Spanish-Ancora</td>
<td>TranKit</td>
<td>99.94</td>
<td>99.13</td>
<td>99.93</td>
<td>99.02</td>
<td>98.94</td>
<td>98.8</td>
<td>99.17</td>
<td>94.11</td>
</tr>
<tr>
<td></td>
<td>Stanza</td>
<td>99.98</td>
<td>99.07</td>
<td>99.98</td>
<td>98.78</td>
<td>98.67</td>
<td>98.59</td>
<td>99.19</td>
<td>92.21</td>
</tr>
<tr>
<td></td>
<td>UDPipe</td>
<td>99.97</td>
<td>98.32</td>
<td>99.95</td>
<td>98.32</td>
<td>98.13</td>
<td>98.13</td>
<td>98.48</td>
<td>88.22</td>
</tr>
<tr>
<td></td>
<td>spaCy</td>
<td>99.95</td>
<td>97.54</td>
<td>99.43</td>
<td>93.43</td>
<td>-</td>
<td>-</td>
<td>80.02</td>
<td>89.35</td>
</tr>
</tbody>
</table>

# System Evaluation

- Performance comparison on NER test sets:

<table>
<thead>
<tr>
<th>Language</th>
<th>Corpus</th>
<th>Trankit</th>
<th>Stanza</th>
<th>Flair</th>
<th>spaCy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>AQMAR</td>
<td>74.8</td>
<td>74.3</td>
<td>74.0</td>
<td>-</td>
</tr>
<tr>
<td>Chinese</td>
<td>OntoNotes</td>
<td>80.0</td>
<td>79.2</td>
<td>-</td>
<td>69.3</td>
</tr>
<tr>
<td>Dutch</td>
<td>CoNLL02</td>
<td>91.8</td>
<td>89.2</td>
<td>91.3</td>
<td>73.8</td>
</tr>
<tr>
<td></td>
<td>WikiNER</td>
<td>94.8</td>
<td>94.8</td>
<td>94.8</td>
<td>90.9</td>
</tr>
<tr>
<td>English</td>
<td>CoNLL03</td>
<td>92.1</td>
<td>92.1</td>
<td>92.7</td>
<td>81.0</td>
</tr>
<tr>
<td></td>
<td>OntoNotes</td>
<td>89.6</td>
<td>88.8</td>
<td>89.0</td>
<td>85.4</td>
</tr>
<tr>
<td>French</td>
<td>WikiNER</td>
<td>92.3</td>
<td>92.9</td>
<td>92.5</td>
<td>88.8</td>
</tr>
<tr>
<td>German</td>
<td>CoNLL03</td>
<td>84.6</td>
<td>81.9</td>
<td>82.5</td>
<td>63.9</td>
</tr>
<tr>
<td></td>
<td>GermEval14</td>
<td>86.9</td>
<td>85.2</td>
<td>85.4</td>
<td>68.4</td>
</tr>
<tr>
<td>Russian</td>
<td>WikiNER</td>
<td>92.8</td>
<td>92.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Spanish</td>
<td>CoNLL02</td>
<td>88.9</td>
<td>88.1</td>
<td>87.3</td>
<td>77.5</td>
</tr>
</tbody>
</table>
System Evaluation

- Speed and memory comparison:

<table>
<thead>
<tr>
<th>System</th>
<th>GPU UD</th>
<th>GPU NER</th>
<th>CPU UD</th>
<th>CPU NER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trankit</td>
<td>4.50×</td>
<td>1.36×</td>
<td>19.8×</td>
<td>31.5×</td>
</tr>
<tr>
<td>Stanza</td>
<td>3.22×</td>
<td>1.08×</td>
<td>10.3×</td>
<td>17.7×</td>
</tr>
<tr>
<td>UDPipe</td>
<td>-</td>
<td>-</td>
<td>4.30×</td>
<td>-</td>
</tr>
<tr>
<td>Flair</td>
<td>-</td>
<td>1.17×</td>
<td>-</td>
<td>51.8×</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model Package</th>
<th>Trankit</th>
<th>Stanza</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilingual Transformer</td>
<td>1146.9MB</td>
<td>-</td>
</tr>
<tr>
<td>Arabic</td>
<td>38.6MB</td>
<td>393.9MB</td>
</tr>
<tr>
<td>Chinese</td>
<td>40.6MB</td>
<td>225.2MB</td>
</tr>
<tr>
<td>English</td>
<td>47.9MB</td>
<td>383.5MB</td>
</tr>
<tr>
<td>French</td>
<td>39.6MB</td>
<td>561.9MB</td>
</tr>
<tr>
<td>Spanish</td>
<td>37.3MB</td>
<td>556.1MB</td>
</tr>
<tr>
<td><strong>Total size</strong></td>
<td>1350.9MB</td>
<td>2120.6MB</td>
</tr>
</tbody>
</table>
Usage

- **Installation:** `pip install trankit`
- **Github:** [https://github.com/nlp-uoregon/trankit](https://github.com/nlp-uoregon/trankit)
- **Demo website:** [http://nlp.uoregon.edu/trankit](http://nlp.uoregon.edu/trankit)
Basic functions:

```python
from trankit import Pipeline

# initialize a pipeline on English
p = Pipeline(lang='english', gpu=True, cache_dir='./cache')

doc = '''Michael helped shoot the majority of my firm's website
and we could not have been happier.'''

# perform all tasks on the input
all = p(doc)

sents = p.ssplit(doc)  # sentence segmentation

tokens = p.tokenize(doc)  # tokenization

posdeps = p.posdep(doc)  # upos, xpos, ufeats, dependency parsing

ners = p.ner(doc)  # ner tagging

lemmas = p.lemmatize(doc)  # lemmatization
```
Basic functions:

```javascript
// Output
{
  'text': 'Hello! This is Trankit.', // input string
  'sentences': [ // list of sentences
    {
      'id': 1, 'text': 'Hello!', 'dspan': (0, 6), 'tokens': [...]
    },
    {
      'id': 2, // sentence index
      'text': 'This is Trankit.', 'dspan': (7, 23), // sentence span
      'tokens': [ // list of tokens
        {
          'id': 1, // token index
          'text': 'This', 'upos': 'PRON', 'xpos': 'DT', 'feats': 'Number=Sing|PronType=Dem',
          'head': 3, 'deprel': 'nsubj', 'lemma': 'this', 'ner': 'O',
          'dspan': (7, 11), // document-level span of the token
          'span': (0, 4) // sentence-level span of the token
        },
        {
          'id': 2...
        },
        {
          'id': 3...
        },
        {
          'id': 4...
        }
      ]
    }
  ]
}
```
Usage

- Multilingual case:

```python
from tranltk import Pipeline

# initialize a multilingual pipeline
p = Pipeline(lang='auto')

# Tokenize an English input
en = p.tokenize('Rich was here before the scheduled time.

# Perform POS, Morphological tagging and Dependency parsing for a French input
fr = p.posdep('Toutefois, les filles adorent les desserts.

# Perform NER tagging for a Chinese input
zh = p.ner('可见牧牛图是根据宋神宗时，官拜礼部郎杨次公的“牧牛颂”而创作的。')
```
Usage

- Training your own pipelines:

```python
from tranakit import TPipeline

tp = TPipeline(training_config={
    'task': 'tokenize',
    'save_dir': './saved_model',
    'train_txt_fpath': './train.txt',
    'train_conllu_fpath': './train.conllu',
    'dev_txt_fpath': './dev.txt',
    'dev_conllu_fpath': './dev.conllu'}

trainer.train()
```
Thank you for listening!