به نام خدا

کارگاه Dadmatools درس NLP بهاره کاوسی نژاد – 99431217

- Informal2Formal:

مشاهده می شود که در مثال اول "استاده" و "همش" و در مثال دوم "شمرون"، "معلممون" و "گنده" به حالت رسمی تبدیل نشده اند.

Testing Informal2Formal

```
[19] print(translator.translate('این استاده چرا انگیلیسی حرف می زنه همش'))
```

```
این استاد است چرا انگلیسی حرف می‌زند همه او
```

```
print(translator.translate('دروازه شمرون یه خونه بود که معلممون اونجا یه مغازه گنده داشت'))
```

```
دروازه شمرو هستند یک خونه بود که معلممون آنجا یک مغازه گنده داشت
```

- Named Entity Recognition:

در مثال اول "رفت" و در مثال دوم تمامی کلمات اشتباه تشخیص داده شده اند.

```
[25] import dadmatools.pipeline.language as language

     # as tokenizer is the default tool, it will be loaded even without calling
     pips = 'ner'
     nlp = language.Pipeline(pips)
     text = 'محمد محمدزاده گرشاسبي به استانداري سن ديگو رفت'
     doc = nlp(text)
     doc
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading multi-word expander for persian
Loading NER tagger for persian
================================================
Active language: persian
================================================
{'sentences': [{'id': 1,
   'tokens': [{'id': 1, 'text': 'محمد', 'ner': 'B-PER'},
    {'id': 2, 'text': 'محمدزاده', 'ner': 'I-PER'},
    {'id': 3, 'text': 'گرشاسبي', 'ner': 'E-PER'},
    {'id': 4, 'text': 'به', 'ner': 'O'},
    {'id': 5, 'text': 'استانداري', 'ner': 'B-ORG'},
    {'id': 6, 'text': 'سن', 'ner': 'I-ORG'},
    {'id': 7, 'text': 'ديگو', 'ner': 'E-ORG'},
    {'id': 8, 'text': 'رفت', 'ner': 'O'}]}],
 'lang': 'persian'}
```

```python
# as tokenizer is the default tool, it will be loaded even without calling
pips = 'ner'
nlp = language.Pipeline(pips)
text = 'اي پيك نسيم صبا بار دگر گر به سر کوی دوست بگذري'
doc = nlp(text)
doc
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading multi-word expander for persian
Loading NER tagger for persian
================================================
Active language: persian
================================================
{'sentences': [{'id': 1,
   'tokens': [{'id': 1, 'text': 'اي', 'ner': 'O'},
    {'id': 2, 'text': 'پيك', 'ner': 'O'},
    {'id': 3, 'text': 'نسيم', 'ner': 'O'},
    {'id': 4, 'text': 'صبا', 'ner': 'O'},
    {'id': 5, 'text': 'بار', 'ner': 'O'},
    {'id': 6, 'text': 'دگر', 'ner': 'O'},
    {'id': 7, 'text': 'گر', 'ner': 'O'},
    {'id': 8, 'text': 'به', 'ner': 'O'},
    {'id': 9, 'text': 'سر', 'ner': 'O'},
    {'id': 10, 'text': 'کوی', 'ner': 'O'},
    {'id': 11, 'text': 'دوست', 'ner': 'O'},
    {'id': 12, 'text': 'بگذري', 'ner': 'O'}]}],
 'lang': 'persian'}
```

- Part of Speech Tagging:

در این مثال "بگذری" اسم شناخته شده است درحالیکه فعل است.

Testing Part of Speech Tagging

```python
# as tokenizer is the default tool, it will be loaded even without calling
pips = 'pos'
nlp = language.Pipeline(pips)
text = 'اي پيك نسيم صبا بار دگر گر به سر کوی دوست بگذري'
doc = nlp(text)
doc
```

```
    'head': 8,
    'deprel': 'nmod:poss'},
   {'id': 5,
    'text': 'بار',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 8,
    'deprel': 'nmod:poss'},
   {'id': 6,
    'text': 'دگر',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 8,
    'deprel': 'nmod:poss'},
   {'id': 7,
    'text': 'گر',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 8,
    'deprel': 'nmod:poss'}
```

```
        {'id': 9,
         'text': 'سر',
         'upos': 'NOUN',
         'xpos': 'N_SING',
         'feats': 'Number=Sing',
         'head': 8,
         'deprel': 'obj'},
        {'id': 10,
         'text': 'کوی',
         'upos': 'NOUN',
         'xpos': 'N_SING',
         'feats': 'Number=Sing',
         'head': 8,
         'deprel': 'nmod:poss'},
        {'id': 11,
         'text': 'دوست',
         'upos': 'NOUN',
         'xpos': 'N_SING',
         'feats': 'Number=Sing',
         'head': 8,
         'deprel': 'nsubj'},
        {'id': 12,
         'text': 'بگذري',
         'upos': 'NOUN',
         'xpos': 'N_SING',
         'feats': 'Number=Sing',
         'head': 8,
         'deprel': 'root'}]}],
    'lang': 'persian'}
```

- Dependency Parsing:

```
[28] # as tokenizer is the default tool, it will be loaded even without calling
     pips = 'dep'
     nlp = language.Pipeline(pips)
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading tagger for persian
Loading multi-word expander for persian
================================================
Active language: persian
================================================
```

```
[29] text = 'مرده آن است که نبرند نامش به نکویی'
     doc = nlp(text)
     doc
```

```
{'sentences': [{'id': 1,
   'tokens': [{'id': 1,
     'text': 'مرده',
     'upos': 'NOUN',
     'xpos': 'N_SING',
     'feats': 'Number=Sing',
     'head': 0,
     'deprel': 'root'},
    'upos': 'SCONJ',
    'xpos': 'CON',
    'head': 2,
    'deprel': 'mark'},
   {'id': 5,
    'text': 'نبرند',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 2,
    'deprel': 'ccomp'},
   {'id': 6,
    'text': 'نامش',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 2,
    'deprel': 'nsubj'},
   {'id': 7,
    'text': 'به',
    'upos': 'NOUN',
    'xpos': 'N_SING',
    'feats': 'Number=Sing',
    'head': 2,
```

- Kasreh Ezafe Detection:

در این مثال سپر کسره اضافه ندارد اما تشخیص داده شده است.

Testing Kasreh Ezafe Detection

```
[30] # as tokenizer is the default tool, it will be loaded even without calling
     pips = 'kasreh'
     nlp = language.Pipeline(pips)
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading multi-word expander for persian
Loading Kasreh tagger for persian
================================================
Active language: persian
================================================
```

```
text = 'سپر عقب ماشین جلویی خورد به سپر جلوی ماشین عقبی'
doc = nlp(text)
doc
```

```
{'sentences': [{'id': 1,
   'tokens': [{'id': 1, 'text': 'سپر', 'kasreh': 'S-kasreh'},
    {'id': 2, 'text': 'عقب', 'kasreh': 'O'},
    {'id': 3, 'text': 'ماشین', 'kasreh': 'O'},
    {'id': 4, 'text': 'جلویی', 'kasreh': 'O'},
    {'id': 5, 'text': 'خورد', 'kasreh': 'O'},
    {'id': 6, 'text': 'به', 'kasreh': 'O'},
    {'id': 7, 'text': 'سپر', 'kasreh': 'S-kasreh'},
    {'id': 8, 'text': 'جلوی', 'kasreh': 'O'},
    {'id': 9, 'text': 'ماشین', 'kasreh': 'O'},
    {'id': 10, 'text': 'عقبی', 'kasreh': 'O'}]}],
 'lang': 'persian'}
```

- Spell Checker:

املای کلمات "آمیتیس" و "معطلی" تصحیح نشده است.

```
[32] # as tokenizer is the default tool, it will be loaded even without calling
     pips = 'spellchecker'
     nlp = language.Pipeline(pips)
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading multi-word expander for persian
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_downl
  warnings.warn(
================================================
Active language: persian
================================================
```

```
[33] text = ' آمیتیص پس از مدطي معتلي بلخره رسید '
     doc = nlp(text)
     doc
```

```
1it [00:00,  2.55it/s]
{'spellchecker': {'orginal': ' آمیتیص پس از مدطي معتلي بلخره رسید ',
  'corrected': 'آمیتیص پس از مدتی معتلي بالاخره رسید',
  'checked_words': [('مدتی' ,'مدطي'), ('بالاخره' ,'بلخره'), ('رسید' ,'رسید')]},
 'sentences': [{'id': 1,
   'tokens': [{'id': 1, 'text': 'آمیتیص'},
    {'id': 2, 'text': 'پس'},
    {'id': 3, 'text': 'از'},
    {'id': 4, 'text': 'مدطي'},
    {'id': 5, 'text': 'معتلي'},
    {'id': 6, 'text': 'بلخره'},
    {'id': 7, 'text': 'رسید'}]}],
 'lang': 'persian'}
```

- Normalizer
- Tokenizer:
- Lemmatizer:

"آرد#آ" ریشه درستی نیست.

```
[36] pips = 'lem'
     nlp = language.Pipeline(pips)
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading tagger for persian
Loading multi-word expander for persian
Loading lemmatizer for persian
==================================================
Active language: persian
==================================================
```

```
[38] text = 'گـرد آوري کـردیـد'
     doc = nlp(text)
     doc
```

```
{'sentences': [{'id': 1,
   'tokens': [{'id': 1,
     'text': 'گردآوري',
     'upos': 'NOUN',
     'xpos': 'N_SING',
     'feats': 'Number=Sing',
     'head': 2,
     'deprel': 'root',
     'lemma': 'گردآوري'},
    {'id': 2,
     'text': 'کردید',
     'upos': 'VERB',
     'xpos': 'V_PA',
     'head': 0,
     'deprel': 'root',
     'lemma': 'آ#آرد'}]}],
 'lang': 'persian'}
```

- Sentiment Analysis:

این کنایه معنی منفی ندارد.

Testing Sentiment Analysis

```
[34] pips = 'sent'
     nlp = language.Pipeline(pips)
```

```
Loading pretrained XLM-Roberta, this may take a while...
Model fa_tokenizer exists in cache/dadmatools/fa_tokenizer.pt
Loading tokenizer for persian
Loading multi-word expander for persian
================================================
Active language: persian
================================================
```

```
[35] text = 'آب توی دلش تکان نمی خورد'
     doc = nlp(text)
     doc
```

```
{'sentences': [{'id': 1,
   'tokens': [{'id': 1, 'text': 'آب'},
    {'id': 2, 'text': 'توی'},
    {'id': 3, 'text': 'دلش'},
    {'id': 4, 'text': 'تکان'},
    {'id': 5, 'text': 'نمی'},
    {'id': 6, 'text': 'خورد'}]}],
 'lang': 'persian',
 'sentiment': [{'label': 'negative', 'score': 0.527174711227417}]}
```