TP2 para SPLN

J.João Almeida e Filipe Cunha

Contents

1 Sentiment analisys for PT

Partindo do código

- vader (codigo python, lexicons)
- vader for ptbr
- tradução automática dos lexicons
- alguns recursos cooperativos deversos

fazer um módulo e scripts VADER-UMPT

`https://github.com/cjhutto/vaderSentiment/` `(en)`
`https://github.com/rafjaa/LeIA/` `(ptbr)`

2 odf, docx, libreoffice para transcrição de entrevisas

- doc++ = docx/odf + convenções + marcar com cores. Exemplo
  - person vermelho
  - places verde
  - anotações, comentários (?)
  - elementos com "[ ]", "{ }"
- doc++ do libreoffice para XML
- using odfpy módulo
- using pandoc and pandoc module for python
- odf2xml xml2odf. Exemplo: odf → xml → anotar com spacy → odf

3 serviço remoto de transcrição whisper

- Whisper (openAI, ou sua vertente C++) + modelo hugginface (large-v2)
- P submete audio A
- servidor:
  - guarda A
  - regista pedido numa queue
  - avisa de que vai demorar (estima o tempo necessário), e diz onde vai ficar o resultado final
- quando termina o trabalho, coloca-o no local combinado
- prever um instalador (?)

4 idem para um serviço filtro-linha-de-comando genérico

Exemplo de serviço (transcrição, conversor, . . . , pdf2texto++)

- DSL

conversor(IN,OUT,ERR)= pdftotext IN -o OUT

5 criação de word-embedding multilingue

Dado:

- um dicionário + word-embedding multilingue
- memória de tradução

avaliar a sua qualidade por zonas (para cada zona calcular similaridades e criar um gráfico de qualidade)

Dada uma memória de tradução,

6 Template multi-file

Exemplo de uso: criar uma árvore flit para um utiliário python

6.1 Exemplo inicial

Considere um exemplo do que podeir ser o template multifile

```
=== meta
name:              // provided or ask
author: JJoao      // provided or default

=== tree

pyproject.toml
{{name}}/
  - __init__.py
  - {{name}}.md
exemplo/
README.md
tests/
  - test-1.py

=== pyproject.toml
[build-system]
requires = ["flit_core >=3.2,<4"]
build-backend = "flit_core.buildapi"

[project]
name = "{{name}}"
```

```
authors = [ {name = "{{author}}", email = "FIXME"}]
license = {file = "LICENSE"}
dynamic = ["version", "description"]
dependencies = [ ]
readme = "{{name}}.md"

[project.scripts]
## script1 = "{{name}}:main"

=== {{name}}.md

# NAME

{{name}} - FIXME the fantastic module for...

=== __init__.py
""" FIXME: docstring """
__version__ = "0.1.0"

=== _test-1.py
import pytest
import {{name}}

def test_1():
    assert "FIXME" == "FIXME"
```

6.2 Uso : dado um template gerar árvore do projecto

```
mkfstree -v name=mytool templateflit
```

6.3 Ferramenta complementar: dado uma árvore exemplo, gerar template

```
mktemplateskel -v name=myproj module_myproj/   > template
```

7 Anonimização de dados

"If you think you've anonymized your data, you're probably wrong"

A anonimização de dados é um processo que tem como objetivo remover ou ocultar informações pessoais identificáveis dos dados, de forma a garantir a privacidade dos indivíduos e a proteção de dados sensíveis. Instituições como Tribunais ou Hospitais possuem grandes quantidades de informação que pode ser utilizada para vários fins. Tratando-se de informações sensíveis, tais como acórdãos judiciais e relatórios médicos é necessário proceder à anonimização dos mesmos de modo a garantir a segurança e privacidade das pessoas mencionadas nesses documentos.

Neste trabalho pretende-se investigar, desenvolver e implementar (caso de estudo), uma ferramenta que, dado um determinado texto, anónimiza a sua informação.

Dados que devem ser considerados para anonimização:

- Nomes de pessoas, alcunhas e apelidos
- Moradas
- data e lugar de nascimento
- Números pessoais ou fiscais, número de passaporte, número de carta de condução ou qualquer outro documento pessoal
- Endereços de correio eletrónico, endereços web ou endereço de redes sociais
- Nomes de Organizações

No que diz respeito ao método concreto de anonimização apresentamos as seguintes abordagens:

- O nome, alcunha e apelido reais devem ser substituídos pelas correspondentes iniciais intercaladas com ponto final;
- Os endereços de correio electrónico, endereços web e de redes sociais são substituídos pelo tipo de serviços de internet seguido de três pontos. Por exemplo:
  - Endereço de correio electrónico = email...
  - Página web = www...
  - Endereço de rede social = Facebook... ou Twitter...
- Os números de cartão de cidadão, carta de condução ou outros números de identificação pessoal são anonimizados através da utilização de uma palavra para a descrição do documento, seguida de três pontos. Por exemplo:
  - Número de passaporte 123456789 = Passaporte...

A ferramenta desenvolvida deve ser facilmente instalável em outras máquinas.

8 Pré-Processador de Texto

O pré-processamento de texto é uma etapa essencial em várias tarefas de Processamento de Linguagem Natural (PLN). Normalmente envolve a limpeza e transformação dos dados permitindo que estes sejam usados para outros fins, como por exemplo, treinar word embeddings. Esta tarefa é a primeira etapa de várias pipelines de NLP tendo uma grande influencia em todas as tarefas seguintes. Desta forma, pretende-se desenvolver um toolkit the pré-processamento de texto focado na língua Portuguesa.

Este toolkit deve ter as seguintes funcionalidades, entre outras:

- Tokenização (frases, palavras)
- Capitalização Preferencial
- Tratamento de expressões multi-palavra
- Remoção de stop words
- Normalização de palavras (Stem, Lemma, lowercase, etc)
- Remoção de caracteres especiais
- Restauro de acentuação / capitalização
- Tratamento de números
- Spell checker
- Outros

Sugere-se que as funcionalidades implementadas sejam facilmente configuráveis, podendo-se escolher de forma arbitrária que tipo de pré-processamentos se quer realizar. Seria também interessante haver algumas configurações default para tarefas de NLP especificas.

Finalmente, o toolkit desenvolvido deve ser facilmente instalável em outras máquinas.