Advanced Natural Language Processing and Information Retrieval

Answer sentence reranking pipeline

Alessandro Moschitti, Kateryna Tymoshenko, Antonio Uva
Email: moschitti@disi.unitn.it, k.tymoshenko@gmail.com, auva87@gmail.com

Overview

- Software
  - **https://github.com/iKernels/RelTextRank**
  - you may find the installation instructions there
  - We are going to run a simple end-to-end example
    - **No need to try to install anything related to DKPRO**
- We will build shallow structures with the relational links for the WikiQA corpus

WikiQA corpus

- Download the corpus from
  - https://www.microsoft.com/en-us/download/details.aspx?id=52419
- In the root of RelTextRank do the following
  - The command below converts the corpus in a format employed by the pipeline

```
export wikiqa_location=<folder to which you unpacked the WikiQa distribution>
mkdir data/wikiQA
python scripts/converters/wikiqa_convert.py ${wikiqa_location}/WikiQA-train.tsv data/wikiQA/WikiQA-train.questions.txt  data/wikiQA/WikiQA-train.tsv.resultset
python scripts/converters/wikiqa_convert.py ${wikiqa_location}/WikiQA-test.tsv data/wikiQA/WikiQA-test.questions.txt  data/wikiQA/WikiQA-test.tsv.resultset
python scripts/converters/wikiqa_convert.py ${wikiqa_location}/WikiQA-dev.tsv data/wikiQA/WikiQA-dev.questions.txt  data/wikiQA/WikiQA-dev.tsv.resultset
```

Input data format

QID Text

```
Q1 how are glacier caves formed?
Q2 How are the directions of the velocity and force vectors related in a circular motion
Q5 how did apollo creed die
Q6 how long is the term for federal judges
Q7 how a beretta model 21 pistols magazines works
Q9 how a vul works
```

data/wikiQA/WikiQA-train.questions.txt

QID AnsID IR_rank IR_score label text

```
Q1 Q1-D1-0 0 0.00000 false A partly submerged glacier cave on Perito Moreno Glacier .
Q1 Q1-D1-1 0 0.00000 false The ice facade is approximately 60 m high
Q1 Q1-D1-2 0 0.00000 false Ice formations in the Titlis glacier cave
Q1 Q1-D1-3 0 1.00000 true A glacier cave is a cave formed within the ice of a glacier .
```

data/wikiQA/WikiQA-train.tsv.resultset

Generating training data

Complete data:

```
export CLASSPATH=bin/:target/dependency/*:target/classes


java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/WikiQA-
train.questions.txt -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

To make it faster , you can use a subset of examples (will perform wors):

```
export CLASSPATH=bin/:target/dependency/*:target/classes
head -300 data/wikiQA/WikiQA-train.questions.txt > data/wikiQA/train300.questions.txt
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
train300.questions.txt  -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

OUTPUT is in data/examples/wikiqa/svm.train, data/examples/wikiqa/svm.train.relevancy

Generating training data

If there is an exception: rm target/dependency/google-collections-1.0.jar

Complete data:

```
export CLASSPATH=bin/:target/dependency/*:target/classes


java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/WikiQA-
train.questions.txt -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

To make it faster , you can use a subset of examples (will perform wors):

```
export CLASSPATH=bin/:target/dependency/*:target/classes
head -300 data/wikiQA/WikiQA-train.questions.txt > data/wikiQA/train300.questions.txt
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
train300.questions.txt  -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

OUTPUT is in data/examples/wikiqa/svm.train,
data/examples/wikiqa/svm.train.relevancy

Generating training data

Complete data:

```
SSPATH=bi...dependency/*:ta...sses

java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/WikiQA-
train.questions.txt -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train
-expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

Entry point

Candidate answers (APs)

Where to persist the CASes

Questions

Do we TRAIN or TEST?

APs to keep per Q

Output folder

Defines the Experiment (which structure to generate

Additional experiment configuration (whether to add FOCUs info and do focus match)

Defines which features to extract

To make use a subset ... wors)

```
expo...endency/*...
head -...rain.questions.txt ...train300.questions.txt

java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
train300.questions.txt  -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

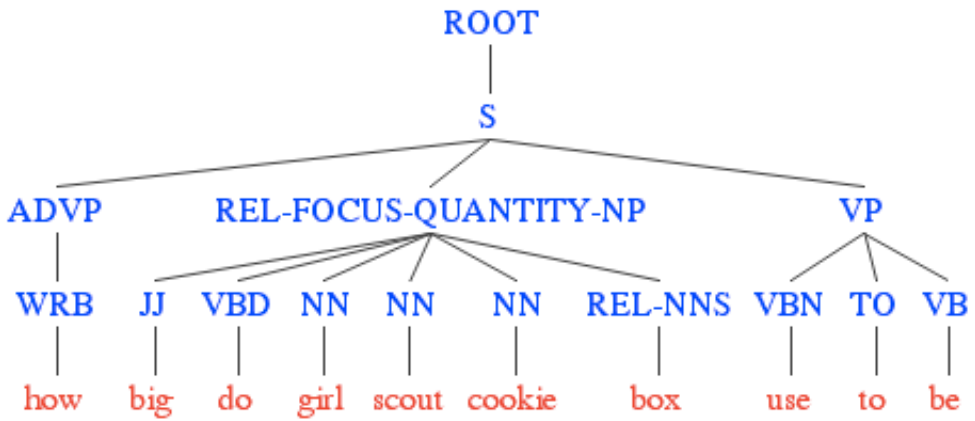
OUTPUT is in data/examples/wikiqa/svm.train, data/examples/wikiqa/svm.train.relevancy

Structure vizualization

You can use ironcreek.net/phpsyntaxtree/? to visualize the structures

how big did girl scout cookie boxes used to be

[ROOT [S [ADVP [WRB [how]]] [REL-FOCUS-QUANTITY-NP [JJ [big]] [VBD [do]] [NN [girl]] [NN [scout]] [NN [cookie]] [REL-NNS [box]]] [VP [VBN [use]] [TO [to]] [VB [be]]]]]



Other `–expClassName` options to try

- it.unitn.nlpir.experiment.fqa.ConstExperiment
- it.unitn.nlpir.experiment.fqa.DT1Experiment
- it.unitn.nlpir.experiment.fqa.DT2Experiment
- it.unitn.nlpir.experiment.fqa.LCTqDT2aExperiment

Generating Development data

Full dev set:

```
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/WikiQA-
dev.questions.txt -answersPath data/wikiQA/WikiQA-dev.tsv.resultset -outputDir data/
examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 1000 -mode test -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

To to test on a subset of dev set:

```
head -50 data/wikiQA/WikiQA-dev.questions.txt > data/wikiQA/dev50.questions.txt
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
dev50.questions.txt -answersPath data/wikiQA/WikiQA-dev.tsv.resultset -outputDir data/
examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 1000 -mode test -
expClassName it.unitn.nlpir.experiment.fqa.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/original.config
```

OUTPUT is in data/examples/wikiqa/svm.test,
data/examples/wikiqa/svm.test

If question classification does not work: backup

Train:

```
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
train300.questions.txt  -answersPath data/wikiQA/WikiQA-train.tsv.resultset -outputDir
data/examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 10 -mode train -
expClassName it.unitn.nlpir.experiment.fqa.nofocqc.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/nofrel.config
```

Test:

```
java -Xmx5G -Xss512m -Dlogback.configurationFile=resources/logback.xml
it.unitn.nlpir.system.core.ClassTextPairConversion -questionsPath data/wikiQA/
dev50.questions.txt -answersPath data/wikiQA/WikiQA-dev.tsv.resultset -outputDir data/
examples/wikiqa -filePersistence CASes/wikiQA -candidatesToKeep 1000 -mode test -
expClassName it.unitn.nlpir.experiment.fqa.nofocqc.CHExperiment -featureExtractorClass
it.unitn.nlpir.features.presets.BaselineFeatures -expConfigPath config/nofrel.config
```

Train and test

Build the tool:

```
cd tools/SVM-Light-1.5-rer
make clean; make
cd ../..
```

Train and test:

```
tools/SVM-Light-1.5-rer/svm_learn -t 5 -F 3 -C + -m 5000  data/examples/wikiqa/
svm.train data/wikiQA/wikiqa-ch-baselinefeats.model
```

```
tools/SVM-Light-1.5-rer/svm_classify data/examples/wikiqa/svm.test data/wikiQA/wikiqa-
ch-baselinefeats.model  data/examples/wikiqa/wikiqa-ch-baselinefeats.pred
```

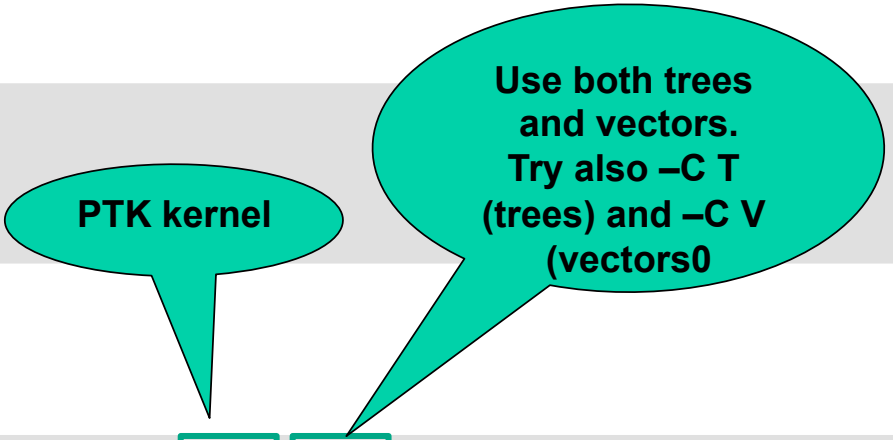
Evaluate:

```
python scripts/eval/ev.py --ignore_noanswer --ignore_allanswer -t 1000 data/examples/
wikiQA/svm.relevancy data/examples/wikiqa/wikiqa-ch-baselinefeats.pred
```

Train and test

Build the tool:

```
cd tools/SVM-Light-1.5-rer
make clean; make
cd ../..
```



Train and test:

```
tools/SVM-Light-1.5-rer/svm_learn -t 5 -F 3 -C + -m 5000  data/examples/wikiqa/
svm.train data/wikiQA/wikiqa-ch-baselinefeats.model  data/examples/wikiqa/wikiqa-ch-
baselinefeats.pred
```

```
tools/SVM-Light-1.5-rer/svm_classify data/examples/wikiqa/svm.test data/wikiQA/wikiqa-
ch-baselinefeats.model  data/examples/wikiqa/wikiqa-ch-baselinefeats.pred
```

Evaluate:

```
python scripts/eval/ev.py --ignore_noanswer --ignore_allanswer -t 1000 data/examples/
wikiQA/svm.relevancy data/examples/wikiqa/wikiqa-ch-baselinefeats.pred
```

All slides are in https://github.com/iKernels/RelTextRank
In in the ainlp slides folder

Please us the instructions on github for installing, and instructions on these slides for running the experiments