NLP and IR

Lab05: Building Feature Vector Representation for STS

Aliaksei Severyn

University of Trento, Italy

May 14, 2013

Plan for the lab

- Modeling input objects
  - Feature-based representation
  - Structural representation
- STS
  - Intro to the task
  - Examples of implemented features
  - In-class experiments

Supervised Learning to-do list

- Get labeled data
- Identify the target task
  - Classification binary/multi-class, reranking, regression, etc.
- Select learning algorithms
  - **SVM**, k-NN, Naïve Bayes, Decision Trees, etc.
- Build representation of input data
  - **Feature vectors**
  - Structural objects, e.g. sequences, trees, graphs

Representation overview

Features	Structures
Provide explicit encoding of input objects into Euclidian space	Map input objects into implicit high-dimensional space (can be infinite-dimensional)
Each feature represents a dimension characterizing the input object. Relatively small number of dimensions	Generate huge number of dimensions
Often require significant engineering effort	Requires selection of a good kernel. Comes for free ones the kernel.
Good prior knowledge of the task and domain is essential to build good features	Is less domain-dependent
Learning is fast	Much slower in training

Semantic Textual Similarity (STS)

- Goal
  - Build a component to compute semantic textual similarity of texts
  - Useful for many NLP tasks: QA, paraphrasing, Textual Entailment, Machine Translation, etc.
- Data
  - Pairs of **short** texts
  - Labels - similarity degree provided by **humans** (5 semantic equivalence to 0 - no relation)
- Our task
  - Build a system able to **automatically** predict similarity scores between document pairs

STS-2012 task description

- Training
  - 3 datasets with texts extracted from MSRPar (paraphrasing), MSRvid (video descriptions), SMT (machine translation)
  - 2234 training examples
- Test
  - 3 datasets (same as in training)
  - 2 (surprise) datasets: OnWN (ontology mappings) and SMTnews (news)
- Evaluation: Pearson correlation w.r.t. human jugdements
- More info about the task:
  - STS-2012 http://www.cs.york.ac.uk/semeval-2012/task6/
  - STS-2013 http://ixa2.si.ehu.es/sts/

STS-2012 approaches

- Most successful systems:
  - Use machine learning methods to learn a scoring function
  - Encode a large variety of various similarity features
- 30 teams competing with 90 systems
- 2 best systems released as open source:
  - Takelab
  - DKPro
- Datasets and software:
    - http://www-nlp.stanford.edu/wiki/STS

STS-2012 Competition

rank	run	ALL	MSRpar	MSRvid	SMT-eur	On-WN	SMT-news
1	baer/task6-UKP-run2_plus_postprocessing_smt_twsi	.8239	.6830	.8739	.5280	.6641	.4937
2	jan_snajder/task6-takelab-syntax	.8138	.6985	.8620	.3612	.7049	.4683
3	jan_snajder/task6-takelab-simple	.8133	.7343	.8803	.4771	.6797	.3989
4	baer/task6-UKP-run1	.8117	.6821	.8708	.5118	.6649	.4672
5	rada/task6-UNT-IndividualRegression	.7846	.5353	.8750	.4203	.6715	.4033
6	mheilman/task6-ETS-PERPphrases	.7834	.6397	.7200	.4850	.7124	.5312
7	mheilman/task6-ETS-PERP	.7808	.6211	.7210	.4722	.7080	.5149
8	baer/task6-UKP-run3_plus_random	.7790	.6830	.8739	.5280	-.0620	-.0520
9	rada/task6-UNT-IndividualDecTree	.7677	.5693	.8688	.4203	.6491	.2256
10	yeh/task6-SRIUBC-SYSTEM2	.7562	.6050	.7939	.4294	.5871	.3366
11	yeh/task6-SRIUBC-SYSTEM1	.7513	.6084	.7458	.4688	.6315	.3994
12	croce/task6-UNITOR-2_REGRESSION_ALL_FEATURES	.7475	.5763	.8217	.5102	.6591	.4713
13	croce/task6-UNITOR-1_REGRESSION_BEST_FEATURES	.7474	.5695	.8217	.5168	.6591	.4713
14	rada/task6-UNT-CombinedRegression	.7418	.5032	.8695	.4797	.6715	.4033

Building an STS system

- Approach
  - Use **Machine Learning** methods to learn a function mapping text pairs to similarity scores
  - Support Vector Regression
- Representation
  - Pairwise similarity features
    - Encode how similar two texts are using lexical, syntactic, semantic information
    - Each feature is a single score

Similarity measures

- Word similarity
  - Ngram Overlap over raw tokens, lemmas, part-of-speech tags, WordNet senses
  - Weighted word overlap
    - Gives more importance to content words
- Knowledge-based similarity
  - WordNet, WikiPedia (ESA), etc.
- Corpus-based similarity
  - Uses LSA, Topic Modeling, etc. to compute similarity in the topic space

Similarity measures

- Designing your features
  - Example feature
    - Ngram Overlap

$$ngo(S_1, S_2) = 2 \cdot \left(\frac{|S_1|}{|S_1 \cap S_2|} + \frac{|S_2|}{|S_1 \cap S_2|} \right)^{-1}$$

  - Study STS-2012 papers for inspiration
  - Use your knowledge from NLPIR class and intuition to come up with good features

Implementing your metrics for STS

- Pull most recent version from GitHub
- Go to:
  - `~/NLPIR/course_projects/sts2012`
- Read README.md

Overview of the STS framework

- Folder **datasets** raw and annotated data (*.dat)
- Folder **system** contains scripts to:
  - Preprocess the data
    - `corpus_utils.py`
  - Generate features
    - Baseline: `takelab_simple_features.py, takelab_main.py`
    - Your features: `nlpir_main.py`
  - Combine different features sets and generate SVM files:
    - `features2svmfile.py`
- Folder **features** contains features per dataset
- Folder **models** contains SVM files
- **SVM-Light-1.5-rer** – SVR binaries

Replicating exps

From your home:

generate simple features

```
python -u system/takelab_main.py
```

generate SVM files

```
python system/features2svmfile.py
```

train/test a model with simple features

```
sh train-test-eval.sh takelab.simple
```

train/test a more advanced baseline model

```
sh train-test-eval.sh baseline
```

generate your features

```
python -u system/nlpir_main.py
```

Summary

- Feature vector representation in ML
  - Often the most important step for building accurate models
- STS task
  - Description
  - Framework for testing your features
  - Experiments
  - Implementing your own features

References

TakeLab: **Systems for Measuring Semantic Text Similarity**, Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder and Bojana Dalbelo Bašić, Semeval 2012

UKP: **Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures**, Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch, Semeval 2012