Computer Science & Engineering Department

Sentence selection with neural networks over string kernels

Mihai Dan Mașala, Ștefan Rușeți, Traian Rebedea

KES 2017

University POLITEHNICA of Bucharest

Introduction

- Sentence selection: given a question, select the sentence containing the answer, given a list of candidate sentences
- We evaluated an unsupervised method (string kernels) for comparing the question and the answer over two distinct datasets:
  - SQuAD
  - InsuranceQA
- We trained a neural network over different string kernels to try to improve performance

Datasets - SQuAD

- Stanford Question Answering Dataset – SQuAD
- Machine comprehension dataset
- Contains 536 articles and over 100,000 questions about those articles.
- For each question, the answer is represented as a span of text, ranging from a few words to a couple of sentences.
- We adapted this dataset for the sentence selection task
- Very rare cases where answers spans on more than one sentence => Ignored

Datasets - SQuAD

Question: What businesses were the dominant brewers of beer in England by the close of the 17th century?

Traditional English ale was made solely from fermented malt. The practice of adding hops to produce beer was introduced from the Netherlands in the early 15th century. Alehouses would each brew their own distinctive ale , but independent breweries began to appear in the late 17th century. By the end of the century almost all beer was brewed by commercial breweries.

Datasets - InsuranceQA

- Contains over 12.000 questions from the insurance domain.
- A large pool of possible answers is provided
- For each question, the negative examples are randomly selected from the pool
- 500 candidate examples for each question
- Multiple correct answers
- Answers are much longer than the SQuAD ones, containing more sentences

String kernels

Spectrum: $k_p(s,t) = \sum_{v \in \sum p} num_v(s) \cdot num_v(t)$

Intersection: $k_p^{0/1}(s,t) = \sum_{v \in \sum p} in_v(s) \cdot in_v(t)$

Presence: $k_p^{\cap}(s,t) = \sum_{v \in \sum p} \min(num_v(s), num_v(t))$

Where:

$\sum p$ = all p-grams of a given size

$num_v(s)$ = number of occurrences of string (**n-gram**) v as substring in document

$in_v(s)$ = 1 if string (**n-gram**) v occurs as substring in document s, 0 otherwise

Neural networks using string kernels

- We want to find a combination of string kernels that better captures the measure of similarity between the question and the answer.
- From each type of kernel (e.g. intersection, spectrum, presence) we extracted a feature vector based on different n-gram lengths
- Length ranges used: 1-2, 3-4, 5-6, 7-8, 9-10.
- The feature vectors used for training was obtained by concatenating these three vectors.

Neural networks using string kernels

- We then trained a simple feed-forward MLP with one hidden layer, which computes a score for each candidate answer.
- A training example consisted in triplet:
  - Question
  - Correct answer
  - Wrong answer
- Hinge loss:

$$e(q, s^+, s^-) = \max(0, M + sim(q, s^-) - sim(q, s^+))$$

Neural networks using string kernels

Advantages of the model:

- Language independent
- Doesn’t require any external NLP tools
- Doesn’t require a large training set
- Fast training

Disadvantages of the model:

- Simplicity: using only lexical features
- Poor performance compared with the complex deep-learning models on large datasets

Lexical and semantic information

- We added word-level information to see if it brings any improvement
- Lexical information: bag-of-words approach and cosine similarity for comparing the question and answer vectors.
- Semantic information: we used a method based on the average Word2Vec representation of the words in the question and the answer, compared with the cosine similarity
- The similarity computed with the lexical/semantic method is added as a separate feature for the neural network

Results - InsuranceQA

Method	V1 test set 1 (%)	V1 test set 2 (%)	V2 test set(%)
Bag-of-words	49.4	47.0	15.8
Word2Vec bag-of-words	39.2	37.6	10.5
Intersection kernel	51.6	48.5	21.9
Presence kernel	51.7	48.6	21.9
Spectrum kernel	5.1	3.2	2.5
Logistic regression using string kernels	42.3	41.2	15.2
Neural network using string kernels	54.1	50.2	25.0
Neural network using string kernels and semantic information	52.4	51.7	25.4
Neural network using string kernels and lexical information	63.2	56.9	28.2
AP-BILSTM	71.7	66.4	N\A

Results - SQuAD

Method	Dev set (%)
N-gram overlap	74.8
Neural network using Word2Vec average representation	63.8
Intersection kernel	79.5
Presence kernel	79.5
Spectrum kernel	45.0
Neural network using string kernels	81.0
Neural network using string kernels and lexical information	81.7
Neural network using string kernels and semantic information	82.2

Conclusions

- We compared unsupervised and supervised methods based on string kernels for the sentence selection task.
- Among the unsupervised methods, the standalone string kernels obtained the best result on the two QA datasets used.
- We have shown that a simple neural network used on top can improve the performance of string kernels, but it is less effective than deep-learning models with attention for the same task.
- String kernels can be used as a simple standalone comparison technique, or as separate features in a complex model