FACULTY
OF INFORMATICS
Masaryk University

Static Attention-Based Siamese CNNs for Recognizing Faceted Entailment

SUI

Martin Víta

Problem Introduction

Introductory Definitions of Textual Entailment (TE)

- "By *textual entailment* is understood a relationship between coherent text *T* and a language expression *H*, which is considered as a hypothesis. *T entails H* if the meaning of *H* as interpreted in context of *T*, can be deduced from the meaning of *T*."
- "*Textual entailment* is an asymetric relation between two text fragments that describes whether one fragment can be inferred from the other."

If *T* entails *H*, we usually write $T \longrightarrow H$.

Problem Introduction – cont.

Recognizing Textual Entailment (RTE) Task

Recognizing Textual Entailment (RTE) is a binary decision task: whether a given (coherent) text T entails a given text H (in this context often called hypothesis).

If $T \not\rightarrow H$, there is no way how to measure how close is H to some H' such that $T \rightarrow H'$.

In other words, from the RTE viewpoint, a hypothesis completely unrelated to the text T is handled in the same way as a hypothesis that is “almost entailed”.

Approaches to RTE

Basic classification

- (Former) approaches dealing with sequences of words: bag-of-words methods, vector space based models
- “Advanced” approaches: logic based, syntactic-similarity based, decoding methods
- Current approaches: ML based – nowadays, mainly deep learning approaches

Applications of RTE

Why RTE?

– Potentially Myriads of Applications. . .

- Paraphrase detection
- Machine translation evaluation
- Plagiarism detection
- Computing semantic similarity

Annotated Corpora

Notable datasets

- Boeing-Princeton suite
- SNLI: https://nlp.stanford.edu/projects/snli/
- SemEval challenges

Tasks derived from RTE

Partial Textual Entailment

An ordered pair $(T; H)$ forms a partial textual entailment (abbr. as PTE) if *a fragment* of the hypothesis H is entailed by T.

Remark: In this definition, the fragment of the hypothesis is no more defined. Hence, the key question is how to decompose the hypothesis into fragments.

Problem Introduction

Facets: Special Types of Fragments

A facet is an ordered pair of words (f_1, f_2) that are contained in the hypothesis – accompanied by a semantic relation binding these words together. A simplified version of this approach – used in SemEval 2013 Challenge 7 – deals only with a pair of words *without* the semantic relation mentioned explicitly.
This annotated corpus was constructed from students' test responses in biosciences.

For example, if the hypothesis has the form of a sentence "The water was evaporated, leaving the salt.", one of corresponding facets is (evaporated, water), other (leaving, salt) etc.

Remark/Example

An item taken from SciEnts Bank

QUESTION: *You used several methods to separate and identify the substances in mock rocks. How did you separate the salt from the water?*
STUDENT ANSWER: *Let the water evaporate and the salt is left behind.*
REFERENCE ANSWER: *The water was evaporated, leaving the salt.*
FACET: *(evaporated, water)*
In this case, the result is “Expressed” (thus the student’s answer can be regarded as partially correct).
In contrast, when student answers “I don’t know.” the facet *(evaporated, water)* is obviously not expressed.

Problem Introduction

Main Task: Recognizing Faceted Textual Entailment

The problem of recognizing faceted entailment can be stated as follows: *"Does the given text T express the same semantic relationship between the words f_1 and f_2 (that form the facet) exhibited in H?"*

That means we have *just 3 inputs:* text, hypothesis and a pair of facets, and one binary output.

Overview of the Idea Behind our Approach

- We need to represent three entities involved
- Pre-trained embeddings are available
- We want to deal with the text and the hypothesis in a similar way – idea of siamese networks
- (Siamese) CNNs for RTE were introduced a couple of years ago (see the graph/diagram)

First Attempt

- Both text and hypothesis will be “represented” via CNNs with shared weights, i. e. siamese convolutional neural networks
- Facet is represented by a vector obtained as a sum of vector representations of all words (usually two) contained in the facet
- Over this “architecture” there are some fully connected layers and sigmoid neuron for decision about entailment.

Input layers. Sentences (text, hypothesis) are sequences of words – let us assume they have the same length l. Each word w is represented by its embedding $e(w)$ in the d_0-dimensional real space. For the purpose of this work, we use GloVe embeddings with dimension $d_0 = 50$. This leads to a real matrix representation of dimension $(d_0 \times l)$ for each sentence – two main input layers.
We have also an auxiliary input layer for facets. These represented by d_0 dimensional vectors that are centroids of facet word embeddings.

Convolution layers. Let $w_1, w_2, \ldots w_l$ be a sequence of words constituting the considered sentence, $f \in \mathbb{N}$ be the filter width and $\mathbf{c}_i \in \mathbb{R}^{fd_0}$, $0 < i < l + f$ the concatenated embeddings of $e(w_{i-f+1}), \ldots e(w_i)$. For $i < 1$ or $i > l$, $e(w_i)$ are set to zero vectors. The representation $\mathbf{r}_i \in \mathbb{R}^{d_1}$ of an f-gram phrase is described as:

$$\mathbf{r}_i = \mathrm{ReLU}(\mathbf{W} \cdot \mathbf{c_i} + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{d_1 \times fd_0}$ are the convolution weights and $\mathbf{b} \in \mathbb{R}^{d_1}$ a bias. (These layers perform 1d convolution "over time").

Max pooling layers. After convolution with filter width f, we perform column-wise maximum over sliding window of f consecutive columns. Application of such max pooling layer allows us to return back to the same dimension ("length") as the feature map before convolution.

Architecture. After the input layers, there simultaneously are several convolution layers: 6 times with filter width 2, 6 times with filter width 3, and 2 times with filter width 4, these are connected to max pooling layers with corresponding widths. These two instances of CNNs with shared weights provide text/hypothesis vector representations.

Architecture – cont.

These two representations as well as the facet representation become an input to two densely connected layers both with 4 ReLU units. The final decision about faceted entailment is done by a single sigmoid unit. We have 3087 trainable parameters in total. Training was performed in 128 epochs using RMSprop.

Improvements – Attention

- The attention mechanism is based on the assumption that different words in the text/hypothesis have different importance from the faceted entailment viewpoint: *words in the text/hypothesis that are similar to those in the facet should be adjusted in the sentence representation*.
- The similarity on the word level is computed using function $\rho(x, y) = 1/(1 + |x - y|)$ where $|.|$ stands for the standard Euclidean distance.
- For each word w_i in the text/hypothesis ($1 \leq i \leq l$ let λ_i be its weight w.r.t. the facet (v_1, v_2) defined by

$$\lambda_i(w_i) = \max\{\rho(e(w_i), e(v_j)) \mid j = 1, 2\}$$

For words in the hypothesis: $\lambda(.) = 1$.

Attention-Based CNN for recognizing faceted entailment will have formally the same architecture as in the previous case, but instead of feeding the input layer by the concatenated column vectors $e(w_1), e(w_2), \ldots e(w_l)$ for the sentence $w_1, w_2, \ldots w_l$, we will use the concatenation of columns $\lambda_1(w_1) \cdot e(w_1), \lambda_2(w_2) \cdot e(w_2), \ldots \lambda_l(w_l) \cdot e(w_l)$, i. e., embedding vectors multiplied by scalars – the weights with respect to a given facet (in the implementation it is done by point-wise tensor multiplication layer).

Since 75% of sentences (texts, hypotheses) have length lower or equal to 22, all sentences were truncated to 22 words with respect to λ ("those with low similarity with the facet words were removed"). These improvements lead to overall accuracy greater than 0.73.

Tuning

- The amount items of training set is relatively small (compared to the number of parameters) – problem of overfitting
- Regularization is needed (we use L_2 regularization at convolutional layers, and drop-out at fully connected layers)
- We used a dirty trick – we add a part of test set to the training set – 2000 of items were randomly chosen to stay in the test set, the rest was moved to the trainig set. This was done three times.

Current results

Train/test set	Accuracy	Recall
TT1	0.845	0.832
TT2	0.844	0.835
TT3	0.842	0.816

Table: Final Results

Current results – without attention

Train/test set	Accuracy	Recall
TT1	0.786	0.779
TT2	0.780	0.769
TT3	0.783	0.771

Table: Results without attention, on the modified test set

Key Problem: Lack of the Training Data

- Relatively big annotated corpus for RTE is available (Stanford)
- If *T* entails *H*, than any facet obtained from *H*, together with *T* and *H* form a positive instance
- Dependency parsers for English are also available
- Problem of negative instances: borderline cases are most valuable

. . . my current work. . .

Overview of Related Tasks

Opportunity for similar approaches – and relations to RFE

- Recognizing Lexical Entailment
- Recognizing Question Entailment

Acknowledgement

Comments, remarks...

Thank you for your ATTENTION.