Structural Alignment for comparison detection

Introduction

- this paper investigates how to use **semi-supervised strategies** to expand a small set of labeled sentences.

- Specifically, it **uses structural alignment** to starts out from a seed set of manually annotated data and find similar unlabeled sentences to which the labels can be projected.

- For comparison detection, adding the found expansion sentences slightly improves over a non-expanded baseline in low-resource settings.

- It is seen that a substantial proportion of reviews (about 10% of sentences) include explicit textual comparisons, eg.

e.g., “X is better than Y”

Introduction

- MOTIVATION:

Since the higher-level **semantic structure** of **comparisons** as they appear in reviews is **clear-cut**, the problem setting could **respond favorably to weakly supervised training strategies** that start out from a seed set of manually annotated data.

Semantic Role-Labeling (SRL):

Starting with a small set of labeled seed sentences, we use structural alignment, which has been successfully applied to SRL, to automatically find and annotate sentences that are similar to these seed sentences as a way to
get more training data.

Introduction

- General Review sentence structure:

two entities that are compared in some aspect

Hypothesis:

- Predicates that appear in a similar syntactic and semantic context will behave similarly with respect to their arguments so that the labels from the seed sentences can be projected to the unlabeled sentences.

- These newly labeled sentences can then be used as additional training data.

Outline of structural alignment

We collect expansion sentences for a predicate p of a seed sentence s with the following steps for every unlabeled sentence u.

1. **Sentence selection**: Consider u iff it contains a predicate compatible with p.
2. **Argument candidate creation**: Get all argument candidates from s and from u.
3. **Alignment scoring**: Score every possible alignment between the two argument candidate sets.
4. Store best-scoring alignment and its score iff at least one role-bearing node is covered.

When all unlabeled sentences have been processed, we choose the k sentences with the high-est alignment similarity scores as expansion sentences for the seed predicate p.

Steps

Sentence Selection:

- Use part of speech (POS) tagging for all adjectives and adverbs in comparative or superlative form

- consider single-word predicates

Steps

Argument Candidate Selection:

Requirement:

To enlarge the set of argument candidates, while keeping the number of candidates manageable so that alignments can be calculated in reasonable time.

Approach:

- use all ancestors of the predicate until the root and their direct descendants, plus all descendants of the predicate itself.

- remove prepositions and conjunctions

- impose a distance limit and exclude numbers and punctuations.

Steps

Argument Candidate Selection:

Requirement:

To enlarge the set of argument candidates, while keeping the number of candidates manageable so that alignments can be calculated in reasonable time.

Approach:

- use all ancestors of the predicate until the root and their direct descendants, plus all descendants of the predicate itself.
- remove prepositions and conjunctions
- impose a distance limit and exclude numbers and punctuations.

Path-Filtered Approach:

Get paths from the predicate to each argument in the labeled sentence and search for the exact same paths (compared by dependency relations) in the unlabeled sentence. All nodes on the path are extracted as candidates. (sometimes fails)

Steps

Alignment Scoring:

The similarity of an alignment between two sentences s and u is the averaged sum of all word
alignment similarities, themselves the averaged sum of different word similarity measures:

$$\text{score}_s(s,u) = \frac{1}{|M|}\sum_{i=1}^{|M|}\frac{1}{|S|}\sum_{j\in S}\text{sim}_j(w_i,\sigma(w_i))$$

where M is the set of candidates on labeled side, wi ∈ M one of these candidates, σ(wi) the candidate on unlabeled side aligned with wi , and
S is the set of similarities to calculate. Unaligned wi receive a word similarity of zero.

Data

Data Set: English camera reviews.

Divide the data into five folds and use one fold as seed data and the rest as test data.

The full seed data contains 342 sentences with 415 predicates. The test data contains 1365 sentences with 1693 predicates.

As the unlabeled expansion data, we use a set of 280.000 camera review sentences from epinions.com.

To calculate vector space similarities we use co-occurrence vectors (symmetric window of 2 words, retain 2000 most frequent dimensions) extracted from a large set of reviews with a total of 40 million tokens.

Above used set includes camera reviews from amazon.com

Process

Retrain the MATE Semantic Role Labeling system on our data.

Use a typical pipeline setting with three classification steps:

- predicate identification,
- argument identification and
- argument classification.

We distinguish three argument types: two entities and one aspect.

Process

Evaluation:

To evaluate whether the found expansion sentences are useful, we

- add the k best expansion sentences per seed predicate to the seed data and train on this expanded corpus.

- We use the test data for evaluation and compare classification performance of training on the expanded seed data with the baseline trained on the seed data only.

Versions Tested

Use two combinations of similarity measures:

- flat similarities only (S = {vs, dep})

- context based similarities (all, S = {vs, neigh, dep, tok, lev, path}).

    (show Table 1 - Pg.278)

- 4 versions of the expansion:

--- PATH-FLAT: path-filtered candidate creation and flat similarities (closest to the original work).

--- DEP-FLAT: dependency-filtered candidate creation and flat similarities.

--- PATH-CONTEXT: path-filtered candidate creation and context similarities.

--- DEP-CONTEXT: dependency-filtered candidate creation and context similarities.

Results

Questions to address:

1. How many seed sentences should be used (varying d)?
2. How many expansion sentences should be used per seed (varying k)?

For PATH-FLAT , DEP-FLAT and PATH-CONTEXT, almost no setting manages to improve over the non-expanded baseline, every added expansion sentence only decreases performance.

For DEP-CONTEXT, in some cases, especially for low values for d there is a small improvement.

(show graphs – Pg. 279)