Semantic Unit Mining and Embedding

Hao Wang, Litian Ma, Shaoshi Ling
Motivation

Framework

Semantic-Unit Mining
  ○ Phrase Collection & Mining
  ○ Named Entity Typing
  ○ Topic Modeling
  ○ Text Tokenization

Semantic Unit Embedding
  ○ Skip-gram Model

Experiments
  ○ Paraphrase Identification
  ○ Text Classification
  ○ Contextual Semantic-unit Similarity

Conclusions & Future Work
Motivation

- Current embedding methods based on single words, we treat semantic unit including phrase, name entity with typing, and topical word as basic unit to handle limitation of unigram based embedding.

<table>
<thead>
<tr>
<th>Single words (unigram)</th>
<th>Semantic unit (n-grams)</th>
<th>Semantic unit (n-grams)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support, vector, machine</td>
<td>Support vector machine</td>
<td>Support vector machine [Topic]</td>
</tr>
<tr>
<td>Can NOT handle phrase, paraphrase and polysemy</td>
<td>Can handle phrase, paraphrase; but NOT polysemy</td>
<td>Can handle phrase, paraphrase and polysemy</td>
</tr>
</tbody>
</table>
Framework

- Phrase Collection
- Phrase Mining
- Named Entity Typing
- Topic Modeling
- Text Tokenization
- Semantic Unit Embedding
- Paraphrase Identification
- Contextual Semantic Unit Similarity
- Text Classification
Phrase Collection & Mining

- AutoPhrase generates most named entities phrase.

- Knowledge Base has more generalized phrase such as verb phrase.

General phrases

- 95,000+ phrases defined in Wiktionary and WordNet \[1\]

Domain-Specific Terms & Named Entities

- Run AutoPhrase\[2\] on specific corpus

---

Discontinuous Phrase

Identification of phrase continuity\textsuperscript{[1]}:

- For phrase A\textsubscript{B}, we search next 5 words after word A to check whether there is word B can match A to compose a phrase.
  - Heuristics: discontinuous phrases are rarely separated by more than 5 tokens

- Compute the frequency vector \([w_1, w_2, w_3, w_4, w_5]\) in Wikipedia, where \(w_i\) is the number of occurrences of A and B in that order with a distance i.

Named Entity Typing

ClusType[1]:

- **Candidate Generation**
  - Phrase mining on POS-tagged corpus.

- **Heterogeneous Graphs Construction**
  - Collect seed entity mention as labels.

- **Relation Phrase Clustering**
  - Estimate type indicator for unlinkable candidate mentions with the proposed type propagation integrated with relation phrase clustering on the constructed graph.

---

**Topic Modeling**

- Topic models group words/phrases into various topics according to their **semantic meanings**, thus should have **discriminative vectors** in semantic space.[1]

- With the favor of **Latent Dirichlet Allocation**[2] (LDA), we assign a latent topic $z_i \in T$ for each word/phrase $w_i$, according to the probability:

$$
\Pr(z_i | w_i, d) \propto \Pr(w_i | z_i) \Pr(z_i | d).
$$


Text Tokenization

Sentence reformatting:

For sentence ...A...B...C...D...

1. If A_B is a continuous phrase, reformat sentence: ...A...B...C...D... → ...A_B...C...D...
2. If C_D is a discontinuous phrase, reformat sentence: ...A_B...C...D... → ...A_B...C_D...C_D...
3. If A_B is a named entity with typing, reformat sentence: ...A_B... → ...A_B:<Type>...
4. Use LDA to assign topic number k to each single/aggregated word: ...A_B... → ...A_B#k...
Embedding Method

- **Word2vec**

  Skip-gram model objective: Maximize the probability of word given context

  \[
  J_{\Theta} = \frac{1}{T} \sum_{t=1}^{T} \sum_{-n \leq j \leq n, j \neq 0} \log p(w_{t+j} | w_t)
  \]

  Semantic unit based embedding objective:

  \[
  \mathcal{L}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \log \text{Pr}(\langle w_{i+c}, z_{i+c} \rangle | \langle w_i, z_i \rangle)
  \]

  Softmax:

  \[
  \text{Pr}(\langle w_c, z_c \rangle | \langle w_i, z_i \rangle) = \frac{\exp(w_c^{z_c} \cdot w_i^{z_i})}{\sum_{(w_c, z_c) \in \langle W, T \rangle} \exp(w_c^{z_c} \cdot w_i^{z_i})}
  \]
Experiments - Paraphrase Identification

- Microsoft Research Paraphrase (MSRP) corpus
- SemEval 2014 Task 1: Semantic relatedness SICK dataset
- Supervised binary classification
  - Sentence representation equals to the addition over all the token embeddings

Results:

<table>
<thead>
<tr>
<th></th>
<th>MSRP Accuracy</th>
<th>MSRP F1</th>
<th>SICK Spearman</th>
<th>SICK Pearson</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip-gram</td>
<td>0.695</td>
<td>0.805</td>
<td>0.7293</td>
<td>0.7038</td>
</tr>
<tr>
<td>Semantic Unit Embedding</td>
<td>0.711</td>
<td>0.811</td>
<td>0.7442</td>
<td>0.7048</td>
</tr>
</tbody>
</table>
Experiments - Text Classification

- **20NewsGroup**
  - About 20,000 documents from 20 different newsgroups.
- **Generate document embeddings for both training and test set, regard document embedding vectors as features and train a linear classifier SVM.**

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip-Gram</td>
<td>75.4</td>
<td>75.1</td>
<td>74.3</td>
</tr>
<tr>
<td>LDA</td>
<td>72.2</td>
<td>70.8</td>
<td>70.7</td>
</tr>
<tr>
<td>Semantic-unit</td>
<td>77.1</td>
<td>76.8</td>
<td>76.2</td>
</tr>
</tbody>
</table>
Experiments - Contextual Semantic Unit Similarity

- We train our embedding vector on DBLP corpus and selected several phrases and find most similar phrases.

Support vector machine

<table>
<thead>
<tr>
<th>Skip-gram</th>
<th>Semantic-Unit Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>vector</td>
<td>SVM</td>
</tr>
<tr>
<td>matrix</td>
<td>discriminative classifiers</td>
</tr>
<tr>
<td>scalar</td>
<td>kernel-based</td>
</tr>
<tr>
<td></td>
<td>classification</td>
</tr>
</tbody>
</table>
Experiments - Contextual Semantic Unit Similarity

- We trained our model on Wikipedia Corpus.
- We selected several example words and find the most similar semantic units of these semantic units in different topics or typing.

<table>
<thead>
<tr>
<th>Semantic Units</th>
<th>Similar Semantic Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>bank#1</td>
<td>stock, insurance_company, investor</td>
</tr>
<tr>
<td>bank#2</td>
<td>river, edge, coast</td>
</tr>
<tr>
<td>Washington:location</td>
<td>Tacoma, Seattle, Oregon</td>
</tr>
<tr>
<td>Washington:person</td>
<td>thomas_jefferson, american_revolution</td>
</tr>
</tbody>
</table>
Conclusion

- We propose a method to chunk the text into semantic-unit based tokens.

- We introduce the framework to learn embedding representation for semantic units.

- Our experiments show that semantic-unit based embedding outperforms word level based embedding on multiple tasks.
Future Work

- Sparsity of the semantic units.
- Advanced embedding method.
- Experiments on more tasks.
Q & A