Exploiting Semantic Proximity for Information Retrieval

Sanjeet Khaitan, Kamal Verma, Rajat Mohanty, Pushpak Bhattacharyya

Computer Science and Engineering
IIT Bombay
Motivation

- The World Wide Web is the largest information store in the world.
- Traditional search engines use keyword based Information Retrieval.
- Meaning of the query is ignored.
- Irrelevant results are displayed on many occasions.
In this work, we Integrate Information Retrieval (IR) techniques with the Natural Language Processing (NLP) techniques. Represent the meaning in documents using some interlingua, e.g. Universal Networking Language (UNL) or Semantically Relatable Sets (SRSs). Search the documents based on the intermediate representation of both the documents and the query. Precision/Relevance of the results are expected to improve. Intermediate representation, also opens the way for Cross-lingual Information Retrieval.
Semantically Relatable Sets

- Represents sentences as a set of ordered word sets which are semantically related.
- E.g., SRSs for the sentence "The man bought a new car in June" is:
  - \{man, bought\}
  - \{bought, car\}
  - \{bought, in, June\}
  - \{new, car\}
  - \{the, man\}
  - \{a, car\}
SRS Based Search
SRS Based Search:
Motivation

- SRSs represent some form of meaning in the sentences.
  - SRSs have been used as an intermediate step for UNL generation.
- Different from chunks or N-grams where the words in a set are not semantically related.
  - Words in SRSs are in semantic proximity.
SRS Based Search Strategy

Relevance of a sentence ‘s’ w.r.t. a query ‘q’ is:

\[ r_q(s) = \frac{\sum_{srs \in q} \text{weight}(srs) \cdot \text{pres}_s(srs)}{\sum_{srs \in q} \text{weight}(srs)} \]

Document relevance is defined as:

\[ R_q(d) = \frac{\sum_{s \in S_d} r_q(s)}{|S_d|} \]
Experimental Setup

- Text Retrieval Conference (TREC) data was used.
- TREC provides the gold standard for query and relevant documents:

<table>
<thead>
<tr>
<th>Query Number</th>
<th>Document-ID</th>
<th>Relevance Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>WSJ911010-0114</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>WSJ911011-0085</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>AP880304-0049</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>AP880304-0192</td>
<td>0</td>
</tr>
</tbody>
</table>

Table: Relevance Judgments in TREC

- We chose 1919 documents and the first 250 queries.
  - Mostly from the AP newswire, Wall Street Journal and the Ziff data.
Experiment Process

- Lucene with search strategy *tf-idf* as the keyword based search engine.
- Used SRS based search on the other hand.
- Compared both the search methods on various parameters.
Shows that SRS search filters out non-relevant documents much more effectively than the keyword based $tf-idf$ search.
Recall Comparison

- tf-idf consistently outperforms the SRS search engine here.
Mean Average Precision (MAP) Comparison

- MAP contains both recall and precision oriented aspects and is also sensitive to entire ranking.
- SRS Search could not perform here because of the low recall.

\[
MAP = \frac{\sum_{r=1}^{R} \left( P(r) \times rel(r) \right)}{R}
\]
Reasons for Poor Recall
Word Divergence

- **Morphological Divergence**
  - Query: “child abuse”
    - Query SRS: (child, abuse)
  - Sentence: “children are abused”
    - Sentence SRS: (children, abused)

- **Synonymy Divergence**
  - Query: “antitrust cases”
    - Query SRS: (antitrust, cases)
  - Sentence: “An antitrust lawsuit was charged today”.
    - Sentence SRS: (antitrust, lawsuit)

- **Hypernymy Divergence**
  - Query has keyword “car”, while the document has keyword “automobile”.

- **Hyponymy Divergence**
  - Query can be “car” whereas the document might contain “minicar”.

Physical Separation Divergence

- **Query:** “antitrust lawsuit”
  - **Query SRS:** (antitrust, lawsuit)

- **Sentence:** “The federal lawsuit represents the largest antitrust action”
  - **Sentence SRSs:** (lawsuit, represents), (represents, action), (antitrust, action)
Other Divergences

- **Query**: “debt rescheduling”
  - Query SRS: (debt, rescheduling)

- **Sentence**: “rescheduling of debt”
  - Sentence SRS: (rescheduling, of, debt)

- **Query**: “polluted water”
  - Query SRS: (polluted, water)

- **Sentence**: “water pollution has increased in the city”
  - Sentence SRS: (water, pollution)
Miscellaneous Problems

- **Noise in data**
  - Results in the incorrect SRS translation. E.g., unexpected *underscores* between the sentences.

- **Incorrect sentence boundary detection**
  - This results in creation of ungrammatical sentences.

- **List of Information**
  - Each entry inside the list is marked as a sentence.
  - These typically, do not form a grammatical or meaningful sentence.

- **Incorrect SRS generation**
  - The current SRS generator gives incorrect SRSs for some sentences.
  - Results in *non-matching* with a query having similar meaning.
Recall Improvement in SRS Search
Solution for Morphological Divergence

- **Stemming**
  - All words in the document and the query SRSs are stemmed before matching.
  - Gets the base form based on WordNet, while keeping the tag of the word unchanged.
    - `children_NN` stemmed to `child_NN`, but `childish_JJ` not stemmed to `child_NN`
Solution for Synonymy/Hypernymy/Hyponymy Problem

- Word Similarity Calculation
  - Using “WordNet::Similarity” Tool
  - Can't calculate while query processing
    - Query processing may take hours!
  - Can't calculate similarity between all word pairs in corpus
    - 50 days problem!
Getting Related Words

- Used WordNet to find out related words for a given word

Algorithm Outline

1. Get synonyms
2. Get hypernyms upto depth 2
3. Get hyponyms upto depth 2
4. Repeat step 1, 2 and 3 for all synonyms
5. All the words are related words

- Found related words for all words in corpus (Nouns and Verbs).
- Calculated similarity between word and their related words.
SRS Augmentation

- Deals with the “Other Divergences” problem.
- Enriches the SRSs in the corpus.
  - Basically adds new SRSs by applying augment rules on existing SRSs.
Sample Rules I

- **Rule:** $(N1, N2) \Rightarrow (N2(J), N1)$
  - **Sentence:** “water pollution”
  - **Sentence SRS:** $(\text{water}_N, \text{pollution}_N)$
  - **Augmented SRS:** $(\text{polluted}_J, \text{water}_N)$
Rule: \((V, N) \Rightarrow (N, V(N))\)

- **Sentence**: “destroy city”
- **Sentence SRS**: \((\text{destroy}_V, \text{city}_N)\)
- **Augmented SRS**: \((\text{city}_N, \text{destruction}_N)\)
Sample Rules III

- Rule: \((N1, of, N2) \Rightarrow (N2, N1)\)
  - Sentence: “rescheduling of debt”
  - Sentence SRS: \((\text{rescheduling}_N, \text{of}, \text{debt}_N)\)
  - Augmented SRS: \((\text{debt}_N, \text{rescheduling}_N)\)

- Rule: \((N1, of, N2) \Rightarrow (N2(J), N1)\)
  - Sentence: “cup of gold”
  - Sentence SRS: \((\text{cup}_N, \text{of}, \text{gold}_N)\)
  - Augmented SRS: \((\text{golden}_J, \text{cup}_N)\)
Sample Rules IV

- Rule: \((V, \text{ for}, N) \Rightarrow (N, V(N))\)
  - Sentence: “applied for a certificate”
  - Sentence SRS: \((\text{applied}_V, \text{ for}, \text{certificate}_N)\)
  - Augmented SRS: \((\text{certificate}_N, \text{application}_N)\)

- Rule: \((J, \text{ for}, N-\text{ANIMATE}) \Rightarrow (N, J(N))\)
  - Sentence: “famous for her painting”
  - Sentence SRS: \((\text{famous}_J, \text{ for}, \text{painting}_N)\)
  - Augmented SRS: \((\text{painting}_N, \text{fame}_N)\)
  
- Sentence: “It is good for John”
  - Sentence SRS: \((\text{good}_J, \text{ for}, \text{John}_N)\)
  - Augmented SRS: \((\text{John}_N, \text{goodness}_N)\)
Getting Derived Forms

- Get the derived form if available from WordNet.
  - Not always available
- Use Porter Stemmer to get derived form.
Getting Derived Form - Using Porter Stemmer

- Let the word be “national_J”. Want the noun form.
- Step 1. Get the stem using Porter
  - “national” -> “nat”
- Step 2. Get all nouns from WordNet which start with “nat”
  - “nature”, “natural”, “nation”, “nationhood”, “native” etc.
- Step 3. Get the words which have the largest lexicographical match with “national”
  - “nation”, “nationhood”
- Choose any one of them
  - “nation_N”
New System Architecture
Sentence Relevance

\[
    r_q(s) = \frac{\sum_{srs \in q} \max_{srs' \in srsid} (weight(srs) \ast \max_{srs' \in s(t(srs, srs'))})}{\sum_{srs \in q} \max_{srs \in srsid} (weight(srs))}
\]

where,

the SRS Similarity \( t() \) is calculated as

\[
    t(srs, srs') = t(cw1, cw1') \ast equal(fw, fw') \ast t(cw2, cw2')
\]

\( t(w1, w2) \) is calculated using the similarity measure discussed.

\( t(cw1, cw1') \) and \( equal(fw, fw') \) become 1 while matching (FW, CW)s and (CW, CW)s respectively.
Retrieval Scheme

- Step 1. Retrieve top 200 documents using *tf-idf*.
- Step 2. Retrieve top 200 documents using *SRS Based strategy*.
- Step 3. Merge the documents.
- Step 5. Display documents with descending relevance order.
Enhanced SRS search shows huge improvement in recall
Drop in precision noticed but still much higher than *tf-idf*.

Shows that SRS search filters out non-relevant documents much more effectively than the keyword based *tf-idf*. 
Mean Average Precision (MAP) Comparison

- Enhanced SRS Search has been found superior to \textit{tf-idf} on this metric.
- Depicts the overall quality of SRS search.
Results: Discussion

- Recall of the enhanced system improved a lot (0.362 from 0.102)
- Significant rise in MAP (0.149 from 0.054) as well.
- Enhanced SRS based search method dominates \textit{tf-idf} method with
  - High precision (0.131 compared to 0.049).
  - Improved MAP (0.149 compared to 0.86).
- A fall in precision has come into picture because of the boost in recall
  - Still the overall precision is consistently much better than \textit{tf-idf}.
Future Work

- Automatic learning of parameters
  - Weights of SRS
- Physical Separation Divergence Problem needs to be addressed.
Thank You
Cutoff can be varied to set trade-off between recall and precision.
Number of query returns for varying cut-offs.