**Problem**
Given a question \( q \), rank the candidate sentences in a corpus w.r.t. a scoring function \( s(q, p) \) that measures how likely \( p \) answers \( q \).

**Discriminative IR**

\[
\text{argmax}_{p \in P} s(q, p)
\]

We want \( s \) to be

- Trainable by question/answer pairs
- Decomposable into inner products of sparse vectors \( g(q) \cdot f(p) \)

**Under these conditions IR algorithms can be reused!** Given \( q \), we compute the feature vector that is expected from answers: \( t_\theta(f_Q(q)) \).

Given feature functions \( f_Q / f_P / f_{QP} \):

\[
f_{QP}(q, p) = f_Q(q) \otimes f_P(p)
\]

**Motivation**

Vanilla IR – efficient, results not good

Neural reranking – good results, slow (linear)

Is there a better way to triage the set?

**Feature set**

\[
\begin{align*}
&\{(\text{QWord} \otimes \text{LexAnsType}) \otimes \text{NamedEntityTypes}\} + \\
&\{(\text{QWord} \otimes \text{LexAnsType}) \otimes \text{BagOfWords}\} + \\
&(\text{NamedEntities} \otimes \text{NamedEntities}) + \\
&(\text{NormalizedTfIdf} \otimes \text{BagOfWords})
\end{align*}
\]

**Experiments**

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of questions</th>
<th># of sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC/AQUAINT</td>
<td>2150</td>
<td>23,398,842</td>
</tr>
<tr>
<td>WikiQA/Wikipedia</td>
<td>2118</td>
<td>20,368,761</td>
</tr>
</tbody>
</table>

**Evaluation**

<table>
<thead>
<tr>
<th>System</th>
<th>R@1k train</th>
<th>MAP train</th>
<th>MRR train</th>
<th>R@1k dev</th>
<th>MAP dev</th>
<th>MRR dev</th>
<th>R@1k test</th>
<th>MAP test</th>
<th>MRR test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene</td>
<td>35.47%</td>
<td>38.22%</td>
<td>9.78%</td>
<td>35.47%</td>
<td>38.22%</td>
<td>9.78%</td>
<td>35.47%</td>
<td>38.22%</td>
<td>9.78%</td>
</tr>
<tr>
<td>Yao et al. (2013)</td>
<td>25.88%</td>
<td>45.41%</td>
<td>13.75%</td>
<td>25.88%</td>
<td>45.41%</td>
<td>13.75%</td>
<td>25.88%</td>
<td>45.41%</td>
<td>13.75%</td>
</tr>
<tr>
<td>DiscIR</td>
<td>78.20%</td>
<td>75.15%</td>
<td>17.84%</td>
<td>78.20%</td>
<td>75.15%</td>
<td>17.84%</td>
<td>78.20%</td>
<td>75.15%</td>
<td>17.84%</td>
</tr>
<tr>
<td>WikiQA</td>
<td>24.73%</td>
<td>25.69%</td>
<td>0.58%</td>
<td>24.73%</td>
<td>25.69%</td>
<td>0.58%</td>
<td>24.73%</td>
<td>25.69%</td>
<td>0.58%</td>
</tr>
<tr>
<td>DiscIR</td>
<td>58.79%</td>
<td>60.88%</td>
<td>10.26%</td>
<td>58.79%</td>
<td>60.88%</td>
<td>10.26%</td>
<td>58.79%</td>
<td>60.88%</td>
<td>10.26%</td>
</tr>
</tbody>
</table>

**Results**

* Feature vectors are represented as a set of (key = value, weight) tuples.
* For all \((h_i = v_i, w_i) \in f, (h_i = v_i, w_i) \in g\)