1. Introduction

Named Entity Linking (NEL) is the task of mapping in-text mentions of entities to their corresponding entry in a knowledge base, or NIL, if none exist.

Current approaches have a high computational complexity and data retrieval cost with respect to the number of candidates being processed, making it slow and infeasible for use on long documents, big data and real-time data.

2. Hypothesis

– Correct candidates are either highly popular, or share context with the query document.
– Speed of NEL is a function of the number of candidates.
– In-link score is a measure of entity popularity
– Cosine similarity is a measure of shared context but is slow for text [Indyk and Motwani, 1998]
– Pre-filtering must be fast
– Cosine similarity must be approximated

3. Our NEL System

– Candidate extraction: Look for text (mention strings) that might be entities (3% of run time)
– Candidate generation: Find entities (candidates) that match the mention strings (51% of run time)
– Candidate disambiguation: Work out which candidate matches best (46% of run time)

4. Approximating Cosine Similarity

4.1. Locality Sensitive Hashing (LSH)

Overview:
– Documents become points: bag of words (BOW)
– LSH projects high dimensional points to hashes
– Close points have similar hashes
– Candidate hashes are precomputed and stored
– Document hashes are calculated in real-time

Creating hashes:
– Create random hyperplanes (projection matrix)
– Point above hyperplane ← 1
– Point below hyperplane ← 0

Comparing hashes is lightning fast:
– Count the number of shared bits
– Just POPCNT and XOR CPU commands

5. Pre-filtering Results

5.1. Results on training set

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Time (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (100)</td>
<td>100</td>
<td>87.73</td>
</tr>
<tr>
<td>LSH &gt; 0.53 + top 1 in-link</td>
<td>71.83</td>
<td>87.11</td>
</tr>
<tr>
<td>LSH &gt; 0.54 + top 2 in-link</td>
<td>64.26</td>
<td>87.02</td>
</tr>
<tr>
<td>LSH &gt; 0.54 + top 3 in-link</td>
<td>63.78</td>
<td>87.47</td>
</tr>
<tr>
<td>LSH &gt; 0.55 + top 4 in-link</td>
<td>63.12</td>
<td>87.02</td>
</tr>
<tr>
<td>LSH &gt; 0.54 + top 5 in-link</td>
<td>67.34</td>
<td>87.73</td>
</tr>
<tr>
<td>LSH &gt; 0.55 + top 5 in-link</td>
<td>63.16</td>
<td>87.11</td>
</tr>
</tbody>
</table>

5.2. Results on test set

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Time (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (100)</td>
<td>100</td>
<td>71.34</td>
</tr>
<tr>
<td>LSH &gt; 0.54 + top 5 in-link</td>
<td>66.85</td>
<td>71.07</td>
</tr>
</tbody>
</table>

6. Future work

– LSH in candidate disambiguation
– Bloom filter for whole-document linking
– Evaluate local memory LSH and similarity approximations in other tasks

References