Proper-Noun Entity Recognition and Disambiguation system for short text

Pasquale Giantomaso
Student at DIAG Sapienza
via Ariosto, 25, Roma
pasquale.giantomaso@hotmail.it

ABSTRACT
We present a system for detecting and disambiguating proper-noun entities in short text query. The system functioning is based on a dataset containing a set of entity, corresponding to a subset of Freebase entities and stored in a database. Wikipedia pages are being used in order to detect alias, interconnections between entities and confidence score of detected entities. The purpose is to achieve an F-measure reasonable, compared to a set of query outputs, produced by human.

Keywords
Entity Recognition and disambiguation, proper-noun, Freebase, Wikipedia, query, database, F-measure.

1. INTRODUCTION

1.1 Motivation
The search engines try to focus on semantic search, with the purpose of understanding query and documents not only syntactically, but semantically as well. Recent information retrieval studies have focused on entity-related search.

The objective of Entity Recognition and Disambiguation (ERD) system is to recognize mentions of entities in a given text, disambiguate them, and map them to the entities in a given entity collection or knowledge base. The problem is that the entities may appear as different surface forms, and the ERD system try to reach valid entity interpretations, using the context in which the surface form appears. But in short text an ambiguous surface form may match multiple interpretations.

1.2 Approach
The system presented in this paper is an ERD system that recognizes proper-noun entities contained in a data-set (knowledge base) made up as follows:

- this data-set is a subset of Freebase knowledge base, containing entities corresponding to types: Person, Location, Organization (included Web Sites), TV Show, Movie, Book (included comic book), Product (included Videogame, Auto Model and Software), Event and Characters;
- each entity is characterized by FreebaseID, the Freebase knowledge base id, Name, the entity name, and WikipediaTitle, the title of English Wikipedia page corresponding to this entity;
- the data-set contains 2,286,863 entities.

The table 1 is an example of entities in the data-set. For each query the ERD system produces a set of valid entity linking interpretations. The output has the following format:

<table>
<thead>
<tr>
<th>QueryID</th>
<th>InterpretationSet</th>
<th>PrimaryID</th>
<th>MentionText</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q01</td>
<td>0</td>
<td>/m/0fd4x</td>
<td>total recall</td>
<td>1.0</td>
</tr>
<tr>
<td>Q01</td>
<td>1</td>
<td>/m/0gvrws1</td>
<td>total recall</td>
<td>1.0</td>
</tr>
<tr>
<td>Q01</td>
<td>2</td>
<td>/m/0n3zb6d</td>
<td>total recall</td>
<td>1.0</td>
</tr>
</tbody>
</table>

The system recognizes if the query contains more entity corresponding to the same interpretation. For example if the query is “martha stewart and imclone”, the system return:

<table>
<thead>
<tr>
<th>QueryID</th>
<th>InterpretationSet</th>
<th>PrimaryID</th>
<th>MentionText</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q02</td>
<td>0</td>
<td>/m/01b370</td>
<td>martha stewart</td>
<td>1.0</td>
</tr>
<tr>
<td>Q02</td>
<td>0</td>
<td>/m/0j6rg</td>
<td>imclone</td>
<td>1.0</td>
</tr>
</tbody>
</table>

The system recognizes also the alias. For example if the query is “jfk”, the system return:

<table>
<thead>
<tr>
<th>QueryID</th>
<th>InterpretationSet</th>
<th>PrimaryID</th>
<th>MentionText</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q03</td>
<td>0</td>
<td>/m/09sr0</td>
<td>jfk</td>
<td>1.0</td>
</tr>
<tr>
<td>Q03</td>
<td>1</td>
<td>/m/0d3k14</td>
<td>jfk</td>
<td>1.0</td>
</tr>
</tbody>
</table>

where the first interpretation corresponds to the entity named “JFK” (the movie) and the other corresponds to the entity named “John F. Kennedy”, because “jfk” is an alias for this entity.

Therefore the main goal of the method is:

1. Detect whether a proper name refers to a named entity included in the dictionary (detection).
2. Disambiguates between multiple named entity included in the knowledge base (disambiguation).
Table 1. Example of entities in the data-set

<table>
<thead>
<tr>
<th>Freebase id</th>
<th>Entity Name</th>
<th>Wikipedia Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>/m/03grfl9</td>
<td>&quot;Snow&quot;@en</td>
<td>&quot;/wikipedia/en_title/Snow_2004_film&quot;</td>
</tr>
<tr>
<td>/m/0c4487</td>
<td>&quot;Snow&quot;@en</td>
<td>&quot;/wikipedia/en_title/Snow_Pamuk_novel&quot;</td>
</tr>
<tr>
<td>/m/0cn_w1m</td>
<td>&quot;Snow&quot;@en</td>
<td>&quot;/wikipedia/en_title/Snow_1963_film&quot;</td>
</tr>
</tbody>
</table>

2. IMPLEMENTATION

The entities knowledge base is stored in SQL table, containing Freebase id, entity name and link to Wikipedia page corresponding to the entity.

When the system receives a query as an input, it executes the following operations:

- First it checks if the whole text containing in the query is an entity;
- if no, it splits the text in tokens, and check for n-1 token if there is an entity, where n is the number of tokens. For example:
  - "to be or not to be"
  - the system check:
    - "to be or not to" and "be or not to be";
- if the system recognizes an entity, it checks only the remaining text;
- otherwise it checks for n-2, n-3,...,1 tokens.

Here an example of execution:

Query : “michael jordan basket”

1. The system discovers that the whole query does not match an entity;
2. splits the query in “michael jordan” and “jordan basket” and discovers that “michael jordan” match to several entities;
3. the remaining text is “basket” and the system discovers that does not match an entity.

An example of token division if no entity is recognized is represented in figure 1.

The following SQL query is used to check if the text match to an entity:

```sql
SELECT * FROM erd.kb2 WHERE entity=? or wiki=?
```

Where the first '?' is replaced with the text that the system have to check, and the last '?' is replaced with the Wikipedia link of the page, corresponding to an entity for which the text is recognized as an alias (show section 3.1).

Figure 1. Example of token division

```
<table>
<thead>
<tr>
<th>the atypical squamous cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>the atypical squamous cells</td>
</tr>
<tr>
<td>atypical squamous cells</td>
</tr>
<tr>
<td>the atypical squamous cells</td>
</tr>
<tr>
<td>the</td>
</tr>
<tr>
<td>squamous cells</td>
</tr>
</tbody>
</table>
```

Very important for the English language is the recognize of possessive case. The problem is that in the knowledge base there is entities whose name contains the possessive case. Then the system when find a possessive case, check first if exists an entity corresponding to the text containing the possessive case, if not check if the text, excluding the ‘s, corresponds to an entity.

For example the query “kobe bryant's wife” returns:

```
Q04 0 /m/01kmd4 kobe bryant 1.0
```

because the system recognizes the entity “Kobe Bryant” though there is a possessive case.

2.1 Technical Report

The ERD system was implemented as a web service following a REST based protocol in Java language. The Database Management System used is MySQL and the Driver JDBC allow the connection between DBMS and application. The Java library Jsoup has been used to connect the application with English Wikipedia (see section 3).

Jsoup is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jQuery-like methods.
The web service is accessible through HTTP POST with parameter:
- **TextID**, UTF-8 String represents the id of the query;
- **Text**, UTF-8 String represents the input query of ERD.

The response body is encoded in UTF-8 and the MIME type set to 'text/plain'.

### 3. WIKIPEDIA AS ENCYCLOPEDIC KNOWLEDGE

Wikipedia is a free online encyclopedia written by volunteers, using a wiki software that allows almost anyone to add and change articles. It is a multilingual resource and it is very dynamic, because article about news or events are added within a few days of their occurrence. Each article in Wikipedia is uniquely identified by its title, a sequence of words separated by underscores, with the first word always capitalized. When the name is ambiguous, it is qualified with a parenthetical expression.

The system uses English Wikipedia in order to find alias, interconnections between entities and confidence score.

#### 3.1 Redirect

The redirect page exists for each alternative name that can be used to refer to an entity in Wikipedia. The name is transformed into a title whose article contains a redirect link to the article for that entity. It is possible that an article contains a considerably high number of redirect pages. For example the page United states is reachable from acronyms (U.S.A, U.S, USA, US), Spanish translation (Los Estados Unidos, Estados Unidos), misspelling (Untied_States) and synonyms (Yankee_land).

#### Figure 2. redirect Java method

```java
public static String redirect(String query){
    String s="";
    Elements imports = doc.select("link[href]");
    for (Element link : imports){
        if(link.tagName("link").attr("rel").equals("canonical"))
            s=link.attr("abs:href").toString();
    }
    return s.substring(s.indexOf("en"));
}
```

The redirect operation of Wikipedia is used to recognize an alias. This is accessible through the URL:

```plaintext
“en.wikipedia.org/wiki/Special:Search/"
```

### 3.2 Entity in the same interpretation

In order to find interconnections between entities, the system checks the Wikipedia pages related to them, analyzing the anchor text in the articles. When the system checks couple of Wikipedia articles corresponding to entity recognized, there are three cases:

1. The first article does not contain link to the second and vice versa. Then the two entities are not part of the same interpretation.
2. The first article contains link to the second and vice versa. Then the two entities are in the same interpretation. For example in the query “michael jordan chicago bulls”, two entities are recognized, “Michael Jordan” and “Chicago Bulls”; the Wikipedia pages related to this entities contain one a link to the other. So the output is:
   - Q05  0  /m/054c1  michael jordan  1.0
   - Q05  0  /m/0jm74  chicago bulls  1.0
3. Only one of two articles contains a link to the other. Then only this entity is recognized as entity. For example if the query is “rincon puerto rico”, two entity are recognized, “Rincon” and “Puerto Rico”; but only the Wikipedia page related to the first entity contains a link to the second entity. So the output is:
   - Q06  0  /m/02_zt7  rincon  1.0

#### Figure 3. wikiConnection Java method

```java
public static boolean wikiConnection(String url,String wiki){
    Document doc =Jsoup.connect("http://"+url).get();
    Elements links = doc.select("a[href]");
    for (Element link : links){
        if(link.attr("abs:href").equals("http://"+wiki))
            return true;
    }
    return false;
}
```

The Java method in figure 3 implements this functionality. It return true if the page with URL url contains a link to a page with URL wiki.
3.3 Score and context

The Wikipedia pages related to entities are also used to find the confidence score of entities recognized. The system computes the confidence score analyzing the article, checking if the query token are contained within it. The purpose of confidence score is to understand the context in which the entity recognized is, through the token which constitute the query.

For example if the query is “bowflex power pro”, the entity “Bowflex” is recognized and the token “bowflex”, “power”, “pro” are contained in the article, exactly in the following snippet of the article:

(…)

In January 2004, about 420,000 Bowflex machines were recalled due to mechanical problems. In November 2004, there was a recall of nearly 800,000 (680,000 Power Pro units and 102,000 Ultimate units) Bowflex machines after reports that several models had broken unexpectedly. (…)

Then the confidence score is 1.0 (100%) and the output is:

Q08 0 /m/04cnvy bowflex 1.0

The following function shows how the score is computed:

$$score(e_j) = \sum_{i=1}^{n} p_i / n \quad (1)$$

$$p_i = \begin{cases} 1 & \text{if token \in article} \\ 0 & \text{otherwise} \end{cases}$$

where $e_j$ is the recognized entity, $n$ is the number of token that constitute the query and $p_i$ is a variable which equals 1 if the article corresponding to entity $e_j$ contains the token $i$.

If there is interconnection between entities (section 3.2) the confidence score is 1.0. The system filter only the entity recognized that have at least a score equal to 0.8 (80%).

If the system recognizes more than one interpretation, then if returns as output only the entities with the max score.

For example if the query is “michael jordan basket player career”, the system recognized two entity and two interpretation with a confidence score at least of 0.8:

Q07 0 /m/04rxmc michael jordan 0.8
Q07 1 /m/054c1 michael jordan 1.0

but only the interpretation with the max score is filtered, because this interpretation corresponds better to the query context:

Q07 0 /m/054c1 michael jordan 1.0

The Confidence Java method in figure 4 returns the confidence score for an entity recognized (input query) and his corresponding Wikipedia page (input url). This method opens a buffered reader on the Wikipedia article and check if the query's tokens are mentioned within it.

Figure 4. confidence Java method

```java
public static double confidence(String query, String url)
{
    double num = 0;
    URL u = new URL("http://"+url);
    BufferedReader in = new BufferedReader(new InputStreamReader(u.openStream()));
    String inputLine;
    LinkedList<String> token = new LinkedList<String>;
    StringTokenizer st = new StringTokenizer(query);
    while (st.hasMoreElements()) {
        String s = st.nextToken();
        token.add(s);
    }
    double den = token.size();
    while ((inputLine = in.readLine()) != null) {
        for (int i = 0; i < token.size(); i++) {
            if (inputLine.toLowerCase().contains(token.get(i).toLowerCase())){
                token.remove(i);
                num++;
            }
        }
        if(token.isEmpty()) break;
    } in.close();
    return num/den;
}
```

Also the most frequent word, as articles, conjunction, etc., are considered in the score calculation, because the probability that appear in all articles is very high.

4. EVALUATION

A query can legitimately have more than one interpretation, thus the system are expected to generate multiple interpretations of non-overlapping linked entity mention, semantically compatible with the query text. These interpretations were compared with annotations produced by the majority agreement among 3 human judges (test-set).

The performance of ERD system was evaluated using average F-measure. Specifically, given a query q, with labeled interpretations $\hat{A}$={$\hat{E}_1$, ...,$\hat{E}_m$}, where each interpretation consists of a set of mentioned entities $E$={$e_1$, ...,$e_l$}. There is no segmentation and the ordering information stored in E. Note that if there are two mentions referring to the same entity, it will only appear once in the set $E$. We define the F-measure of a set of hypothesized interpretations $\hat{A}$={$\hat{E}_1$, ...,$\hat{E}_m$} as:
An hypothesized interpretation is counted as correct only if matches all the entities of an interpretation in the reference label set exactly.

4.1 Analysis results

The test-set, created by 3 human judge, is composed by 91 query annotations. Note that if the system produces no output, the performance measures are: average precision = 0.516, average recall = 1.0 and average F-measure = 0.516, because the test-set is composed by about half of query in which no entity should be recognized.

For analysis purpose we have four version of method (baseline). A baseline is a simple method used to show the improvement of system performance measures.

A first method is a system which does not uses redirect and confidence score computation. The results are: average precision = 0.682, average recall = 0.439 and average F-measure = 0.345.

If we insert in this method the redirect computation, in order to discover alias, the performance measure improves as follow: average precision = 0.795, average recall = 0.494, average F-measure = 0.432.

The third method includes the confidence score computation. It has the following functionality: the system returns an interpretation if the score is at least 0.8 for entity composed by single word or if the is at least 0.66 for entity composed by more words. This assumption is based on the fact that a mention composed by more words has greater probability to be recognized as entity. The performance results are: average precision = 0.826, average recall = 0.725, average F-measure = 0.667.

The last method negates the previous assumption, because if the system returns only the entity recognized with confidence score equals at least to 0.8, the performance measure are improved as follow: average precision = 0.804, average recall = 0.802, average F-measure = 0.700. Note that the average precision decreases compared to the previous, but the average recall increases, because there are less false positive that increases the average F-measure.

The figure 5 and the table 2 show how the performance measures change in different methods.

![performance measures chart](image-url)
Table 2. performance measures values

<table>
<thead>
<tr>
<th></th>
<th>No Redirect</th>
<th>No Score</th>
<th>Differentiated Score</th>
<th>Score 0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average Precision</strong></td>
<td>0.682</td>
<td>0.795</td>
<td>0.826</td>
<td>0.804</td>
</tr>
<tr>
<td><strong>Average Recall</strong></td>
<td>0.439</td>
<td>0.494</td>
<td>0.725</td>
<td>0.802</td>
</tr>
<tr>
<td><strong>Average F-measure</strong></td>
<td><strong>0.345</strong></td>
<td>0.432</td>
<td><strong>0.667</strong></td>
<td><strong>0.700</strong></td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

The purpose of this work is to advance the state of the art on entity recognition and disambiguation, given the increasing support required by search engines. The method shown is an ERD system for short text (query), which tries to recognize proper-noun entity and disambiguate them, respect to a knowledge base stored in memory. The English Wikipedia Encyclopedia has been used in order to detect interconnection between entities, alias and confidence score, by which the system tries to understand the query context. Analyzing the performance measures, the results are an average precision of about 80%, an average recall of about 80% and an average F-measure of about 70%, respect to the human judgment. The response latency of the system can be considered acceptable, but it depends significantly on the computational power of the system in which the ERD system run, mainly for the search into the MySQL database, and the transmission rate of internet connection, mainly for the Wikipedia pages analysis.

6. REFERENCES