**Problem 1 (Entities).**

(a) In entity search, an input prefix $e_1, \ldots, e_k$, $x$ with entities $e_i$ and string $x$ is given and we want to have a short list of auto-completion suggestions for a further entity $e_{k+1}$.

(i) Typically entity candidates are ranked by a weighted sum of the three components: similarity, popularity and relatedness. Propose and justify a measure (formula) for each component.

(ii) Give an example for such an input prefix and some entity candidates for auto-completion. Reason which candidates will more likely get a higher score for each measure in your example (no need for calculations).

(b) Explain the differences between NER and NED. Give an example sentence (containing at least three entities) and perform both tasks on it. If possible, give several potential candidates in the NED step.

**Solution.**

(a) (i) Scores:

- similarity score: $\text{sim}(x, e) = Jaccard(x, e) = \frac{|x \cap e|}{|x \cup e|}$, number of characters string $x$ and entity $e$ share with each other divided by the number of characters appearing in one or both of them; the score ranges between $[0, 1]$
- popularity score: $\text{pop}(e) = \frac{Q_e}{Q}$, where $Q_e$ is the set of queries containing entity $e$ and $Q$ is the set of all previous queries; the more often $e$ is part of a query the higher is its score ranging from $[0, 1]$
- relatedness score: $\text{rel}(e_i, e) = \text{npmi}(e_i, e)$, where $e_i$ is an entity in our input prefix and $\text{npmi}$ is the normalized point-wise mutual information. It returns the fraction of how often these entities appear together compared to appearing individually and is normalized to have a value between $[-1, 1]$.

Note: These are just examples, other reasonable and justified formulas are also ok.

(ii) Example prefix: “Messi Ro”, where $e_1 = "$Messi"$ and $x = \"Ro\"$


“Ronaldo” is most similar to the string “Ro”. Both, “Ronaldo” and “Beckham” have high popularity, “Nowitzki” is probably less popular. “Messi” is more related to “Ronaldo” and “Beckham” because they are all football players, whereas “Nowitzki” is a basketball player. “Ronaldo” is probably more related since he and Messi are current players, while the other two are former ones.

(b) NER: identifies entities in text and assign each to a respective category like PERSON, LOCATION, ORGANIZATION, DATE, etc.


NED:

- candidates for “Trump”: “Donald Trump” (American president), “Ivanka Trump” (Donald Trump’s daughter), “Trump Tower” (skyscraper in New York City), ...
- candidates for “Merkel”: “Angela Merkel” (German chancellor), “Max Merkel” (Austrian football player), “Una Merkel” (American actress), ...
- candidates for “Berlin”: “Berlin” (city, capital of Germany), “East Berlin” (soviet sector of Berlin between 1949 and 1990), ...
Problem 2 (RDF & SPARQL).
Consider the following excerpt of an RDF dataset about athletes and their clubs:

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beckham</td>
<td>wasBornIn</td>
<td>Leytonstone</td>
</tr>
<tr>
<td>Paris</td>
<td>locatedIn</td>
<td>France</td>
</tr>
<tr>
<td>BM</td>
<td>type</td>
<td>SportsClub</td>
</tr>
<tr>
<td>Beckham</td>
<td>hasProfession</td>
<td>SoccerPlayer</td>
</tr>
<tr>
<td>Leytonstone</td>
<td>locatedIn</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>BM</td>
<td>headQuartersIn</td>
<td>Munich</td>
</tr>
<tr>
<td>Lahm</td>
<td>playsFor</td>
<td>BM</td>
</tr>
<tr>
<td>Munich</td>
<td>locatedIn</td>
<td>Germany</td>
</tr>
<tr>
<td>PSG</td>
<td>headQuartersIn</td>
<td>Paris</td>
</tr>
<tr>
<td>Beckham</td>
<td>playsFor</td>
<td>PSG</td>
</tr>
<tr>
<td>Lahm</td>
<td>hasProfession</td>
<td>SoccerPlayer</td>
</tr>
<tr>
<td>Nowitzki</td>
<td>wasBornIn</td>
<td>Wuerzburg</td>
</tr>
<tr>
<td>Dallas</td>
<td>locatedIn</td>
<td>United States</td>
</tr>
<tr>
<td>Lahm</td>
<td>wasBornIn</td>
<td>Munich</td>
</tr>
<tr>
<td>PSG</td>
<td>type</td>
<td>SportsClub</td>
</tr>
<tr>
<td>Nowitzki</td>
<td>playsFor</td>
<td>DM</td>
</tr>
<tr>
<td>DM</td>
<td>headQuartersIn</td>
<td>Dallas</td>
</tr>
<tr>
<td>Wuerzburg</td>
<td>locatedIn</td>
<td>Germany</td>
</tr>
<tr>
<td>Nowitzki</td>
<td>hasProfession</td>
<td>BasketballPlayer</td>
</tr>
<tr>
<td>DM</td>
<td>type</td>
<td>SportsClub</td>
</tr>
</tbody>
</table>

Formulate a SPARQL query to retrieve the following information from the given RDF dataset:

(a) Find Nowitzki’s job

(b) Find the home country of Lahm

(c) Retrieve sport clubs together with their location

(d) Identify soccer players who play in their home country

Solution.

(a)  SELECT DISTINCT ?p WHERE {
       Nowitzki hasProfession ?p.
    }

(b)  SELECT DISTINCT ?country WHERE {
       Lahm wasBornIn ?city.
       ?city locatedIn ?country.
    }

(c)  SELECT DISTINCT ?club ?loc WHERE {
       ?club type SportsClub.
       ?club headQuartersIn ?loc.
    }

(d)  SELECT DISTINCT ?p WHERE {
       ?p hasProfession SoccerPlayer.
       ?p playsFor ?club.
    }
Problem 3 (QA using Group Steiner Trees).

We want to answer the following question using Group Steiner Trees:

“Which footballer won the Golden Ball but missed the Golden Boot?”

The following sentences are given:

1. “A great footballer like Messi unfortunately has missed the Golden Boot.”
2. “Ronaldo has won many awards such as the Golden Ball award and the Golden Shoe, also known as the Golden Boot.”
3. “A player like Ronaldo is very popular.”
4. “Lionel Messi has received the Golden Ball.”

(a) Extract all triples from the given sentences by iteratively applying the following pattern: \((X \ldots \text{verb}(+ \text{preposition}) \ldots Y)\). Additionally, extract type information if available in the sentences using the following pattern \((\text{TYPE} \ldots \text{like/such as} \ldots X)\). Note: For this task no stopword removal is performed. However, consider only nouns and entities (like “Golden Ball”) for \(X\) and \(Y\) and for verbs do not include auxiliar verbs, like “is” or “has” for the relation extraction.

(b) Construct a quasi-KG based on your extracted triples. Create a single node for lexically identical subjects/objects, but multiple nodes for relations even though the name of the relation is identical. Also add type edges.

(c) Calculate weights:

(i) For each node, where the node weight of node \(n\) with label \(l\) is defined as:

\[
\text{max}_i(\cos(\text{w2vec}(l), \text{w2vec}(q_i)))
\]

where \(q_i\) is the most similar term from the question; \(\cos\) is the cosine similarity and \(\text{w2vec}\) denotes the word2vec embedding of the term. If \(l\) consists of several terms, such as in the case of entities, then calculate the score for each term individually and take the average of its sum.

Determine cornerstones (nodes with weight > 0.5).

(ii) For each edge between two nodes with labels \(l_1\) and \(l_2\) as follows:

\[
\frac{1}{\text{dist}(l_1, l_2)}
\]

where \(\text{dist}\) is the distance between the two terms in the respective sentence from which the triple has been retrieved (consider ALL words in the sentence for this calculation). Type edges receive a weight of 1. Furthermore, add alignment edges: add an edge if the label of one node is a substring of the label of another node. Assign these edges also a weight of 1.

(d) Perform the GST algorithm manually. The cost is defined by 1-edge weight. Give the top-3 GSTs you have found.

1 You can use this website to find the similarity between two words: http://bionlp-www.utu.fi/wv_demo/. You have to change the model to “English GoogleNews Negative300”.
(e) Perform answer ranking on your top-3 GSTs: answers are removed if they are no entities. Answers are merged if there is an alignment edge between them. The remaining answers are ranked based on the number of GSTs they appear in.

Solution.

(a) The following triples can be extracted:
- (footballer, missed, Golden Boot)
- (Messi, missed, Golden Boot)
- (Ronaldo, won, awards)
- (Ronaldo, won, Golden Ball award)
- (Ronaldo, won, Golden Shoe)
- (Golden Ball award, known as, Golden Boot)
- (Golden Shoe, known as, Golden Boot)
- (Lionel Messi, received, Golden Ball)

Additionally, the following type information:
- (Messi, type, footballer)
- (Ronaldo, type, player)
- (Golden Ball award, type, awards)
- (Golden Shoe, type, awards)

(b) The quasi-KG is depicted in Figure 1. Here the graph is already shown as undirected graph, since later on the directions will not be considered. However, showing a directed graph is also fine for this step.

(c) (i) The quasi-KG with the annotated node weights is depicted in Figure 2. The colors indicate which question word was most similar to the respective node label. The highlighted nodes are the cornerstones.

(ii) The quasi-KG with edge weights can be seen in Figure 3.

(d) In Figure 4 the top-3 GSTs are displayed. Top-1 GST has a total cost of 2.2, there are two top-2 GSTs both with a total cost of 2.5 and there are also two top-3 GSTs which have both cost 2.7.

Note: A valid GST contains at least one cornerstone from each group. Cornerstones in the same group if they are most similar to the same query term/entity (indicated by the colors in Figure 2).
We get the following groups: (footballer, player), (Golden Ball award, Golden Ball), (Golden Shoe, Golden Boot), (won, won, won), (missed).

(e) Only the nodes “Messi”, “Lionel Messi” and “Ronaldo” can be answers (they are entities and no cornerstones). There is an alignment edge between “Messi” and “Lionel Messi” and therefore their appearances are not counted twice. “Messi” appears in top-1, in one of top-2 and in both of top-3 (4 times in total), whereas “Ronaldo” appears in one of top-2 and one of top-3 (2 times in total). Therefore, “Messi” is the returned answer.
Figure 4: top-3 GSTs

6 of 6