Prof. Dr. Andreas Hotho,
M.Sc. Janna Omeliyanenko
Lecture Chair X for Data Science, Universität Würzburg

8. Exercise for "Sprachverarbeitung und Text Mining"

14.01.2022

1 Graph-based Dependency Parsing

1. How do graph-based dependency parsing approaches differ from shift-reduce-parsing approaches in terms of machine learning?

   - graph-based: learning global, features local
   - transistion-based: learning local, features global

2. Given is the following graph in tabular form that contains the nodes A through D. The table entries describe the edges from one node to the other and (if present) their weights.

   Table 1: Graph: f = from, t = to

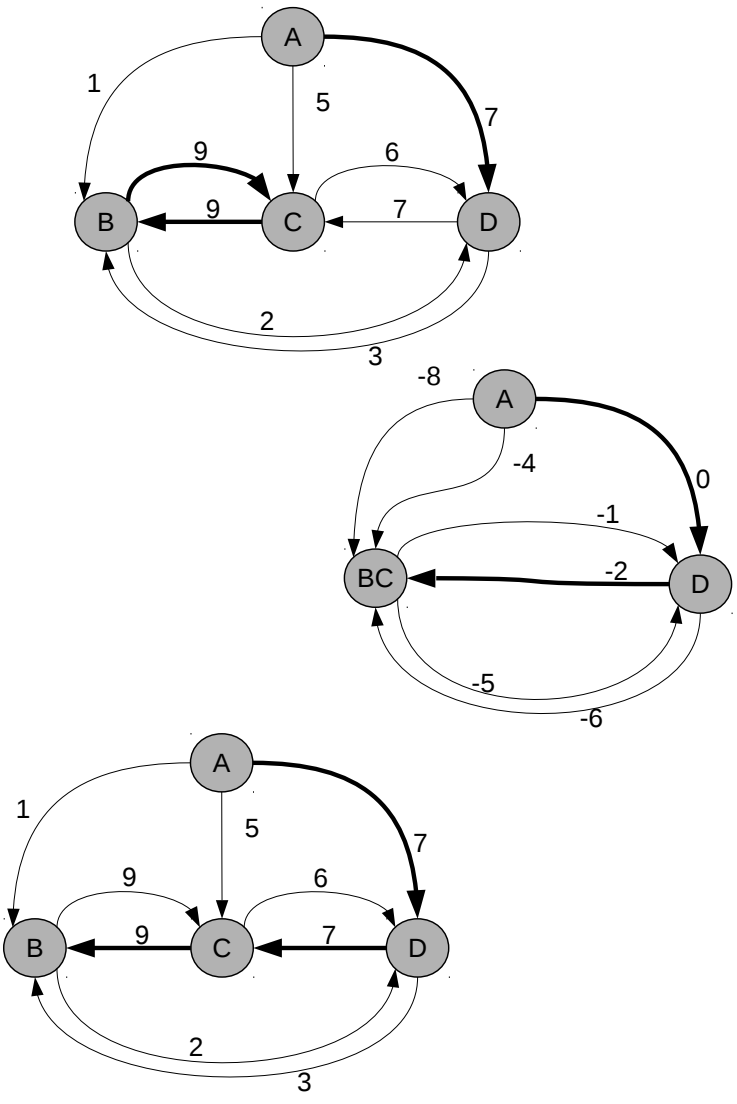
f \ t	A	B	C	D
A	-	1	5	7
B	-	-	9	2
C	-	9	-	6
D	-	3	7	-

   Calculate the maximum spanning tree of the graph using the CLE algorithm.

   - Draw the graph in its original form, mark the initially selected edges.

- For each contracting step, draw the contracted graph, and label the edges with the resulting negative weights. Mark the newly selected edge in the contracted graph.
- Draw the original graph with the newly selected edge instead of the original edge.

For the solution of the task, 3 drawings are required.



2 Named Entity Recognition

1. In the lecture you have learned about BIO and IO encoding for labeling data sets in named entity recognition. Describe a reason for choosing BIO encoding over IO encoding.

   With multiple consecutive I-labels of the same class, it is not possible to distinguish whether the labels belong to one or multiple entities.

2. Define the terms Entity Recognition and Relation Detection in your own words.

   The task of Entity Recognition is to automatically identify and classify entities. An entity here is a sequence of words that describes a real-world entity. The Relation Detection Task determines whether two entities are in relation to each other.

   Which two subtasks can Relation Extraction be divided into?

   - Relation Detection: Recognize existing relationships between 2 entities.
   - Relation Classification: Determine the relation type.

3. Give the following text, write down rules to find all marked Named Entities. Marked entities represent persons(green), locations(blue), and titles(red). You don't need to use a specific syntax. You may assume that the text is fully pre processed. This means you can use POS tags, parse trees, lemmata, etc. as features in your rules.
   An example rule could look as follows:
   NN replied to NN → both NNs are names

   gimli and legolas are walking to mordor. »my father gloin, the ruler of erebor, really enjoyed the last season of 'game of thrones'«, gimli grummeled. » i also enjoyed the part, where dumbledore revealed luke skywalker to be his father.«, legolas replied to gimli. they are joined by aragorn. »who do you think you are, aragorn?«, asks gimli. »i was sent to you by harrys owl.«, he replied.

NN.singular „and“ NN.singular „are“ V → NNs are names
„walking to“ NN → NN is a place
„my father“ NN $ → NN is a name
„ruler of“ NN → NN is a place
„ ′ “ NP „ ′ “ → noun phrase is a title
»S« $,(V NN | NN V | NN V „to” NN) → NNs are names
„joined by” NN.singular → NN is name
NN.singular „revealed” NN.singular NN.singular → names
Possessive-Phrase of NN NN → first NN is a name
»*, NN?« → NN is a name if out-of-language word

3 Non-Structured Hierarchical Classification

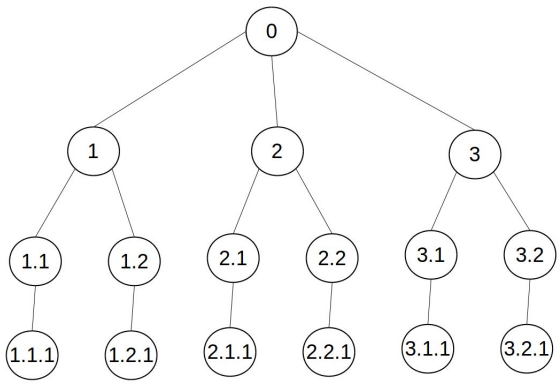
1. Explain in your own words the difference between flat and hierarchical classification.

   In contrast to flat classification, in hierarchical classification the instances can belong to several classes, which in turn are organized hierarchically.

2. What is the potential requirement for classifiers in non-structured hierarchical classification when using a local classifier per parent node?

   The classifiers should be able to perform multiclass classification.

3. Given are the following hierarchy of classifiers and the following training dataset (with a local classifier in each node):



Nr.	Features	Label	S1	S2	S3
1	...	2.2			
2	...	2.1			
3	...	3.2			
4	...	1.2			
5	...	1.1			
6	...	2.2.1			
7	...	2.1.1			
8	...	1.1.1			
9	...	3.1			
10	...	2			
11	...	3			
12	...	1			
13	...	3.2.1			

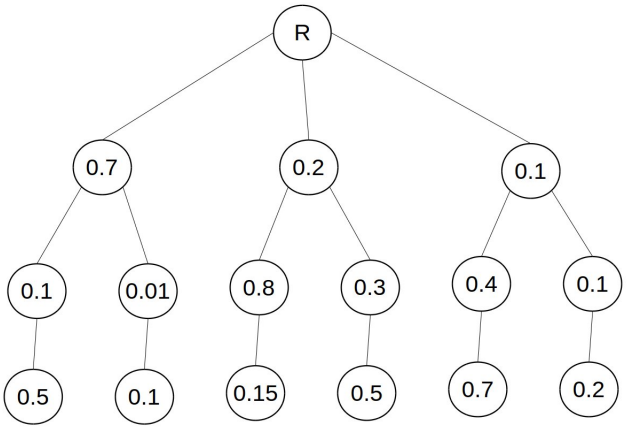
   Consider the classifier 2.2 of the hierarchy. For the following training strategies (S1, S2, S3), indicate which training examples will be used as positive or negative examples, or not used at all.

- S1: Exclusive Policy
- S2: Less-Inclusive Policy
- S3: Siblings Policy

Complete the table by using check marks for positive, crosses for negative, and horizontal lines for non-observed examples.

Nr.	**Features**	**Label**	**S1**	**S2**	**S3**
1	...	2.2	✓	✓	✓
2	...	2.1	x	x	x
3	...	3.2	x	x	-
4	...	1.2	x	x	-
5	...	1.1	x	x	-
6	...	2.2.1	x	✓	✓
7	...	2.1.1	x	x	x
8	...	1.1.1	x	x	-
9	...	3.1	x	x	-
10	...	2	x	x	-
11	...	3	x	x	-
12	...	1	x	x	-
13	...	3.2.1	x	x	-

4. For a given word, the class indicators in the following hierarchy generate the values that are written inside the nodes.



Describe possible problems in the choice of label nodes for the following classification strategies:

- Global Maximum
- Maximum of each layer (top down)

- Global Maximum: 0.8 in layer 2. Problem: hierarchy is never used.
- Maximum of each layer: $[0.7, 0.8, 0.7]$. Problem: Inconsistencies between layers!

What could an improved strategy look like?

Application-dependent, e.g. classes on path with largest sum.

4 Coreference Resolution

1. Define the goal of *Coreference Resolution* in your own words.

   The task of *Coreference Resolution* is to find all mentions that refer to the same entity, and add them to the same cluster.

2. How would you generate training data for the binary classifier in the tournament model out of labeled (coref, non-coref) mention pairs (m_i, m_j)?

Build all possible "fights" and tell the classifier who would win. There are three types of possible fights: coref vs. non coref , coref vs. coref , non coref vs. non coref. First one is easy: Coref wins. In the other cases you need a tie breaker: a possible solution is to choose the pair that has the smaller distance inbetween its mentions.

3. In the most primitive form of this model, there will always be a returned winner. Why is this a problem and how could you solve it?

   If there is a mention that has no coreferent mention, you would introduce a coreference by still picking a winner. You have to introduce a dummy pair with a mention that means "no coreferences" and make the dummy win the tournament.

4. The most primitive tournament type is to let every candidate fight against every other. Which type of tournament would be much more efficient?

   You could do a "King of the hill" tournament. Let the winner of the previous match fight against the next candidate until all possible pairs have fought. The winner of the last match is the winner of the tournament, because he won against all others. You will not get a complete ranking for all pairs, but the winner is all you need for coreference resolution.

5. Calculate all training instances for coreference resolution using the method of Soon et al. using the following sentence:

   $[John]^1_1$ told $[Bill]^2_2$ that $[his]^1_3$ $[mother]^3_4$ was angry.
   $[She]^3_5$ wanted to know where $[he]^1_6$ had been.

   The MentionID is subscript and the EntityID is superscript.

Mention-Pair	coreferent?
(Bill, John)	no
(his, Bill)	no
(his, John)	yes
(mother, his)	no
(mother, Bill)	no
(mother, John)	no
(She, mother)	yes
(he, She)	no
(he, mother)	no
(he, his)	yes