Assignment 3 - Taxonomy Enrichment

1. Task Description

The goal of this assignment is to use methods of distributional semantics, word and node embeddings to solve the taxonomy enrichment task. In the context of this assignment, you will solve one of two tasks at your choice: word sense induction or taxonomy enrichment. Both tasks were formulated earlier (in 2018 and 2020 respectively) in the context of the "Dialogue Evaluation"[1] campaign: RUSSE-2020: Taxonomy Enrichment[2].

You can participate in this task by taking part in the "Taxonomy Enrichment for the Russian Language" competition[3]. All additional information, data and baselines can be found in the repository[4].

https://codalab.lisn.upsaclay.fr/competitions/539

2. Description of the task

Taxonomies are tree structures which organize terms into a semantic hierarchy. Taxonomic relations (or hypernyms) are "is-a" relations: cat is-a animal, banana is-a fruit, Microsoft is-a company, etc. This type of relations is useful in a wide range of natural language processing tasks for performing semantic analysis. The goal of this semantic task is to extend an existing taxonomy with relations of previously unseen words.

Multiple evaluation campaigns for hypernym extraction (SemEval-2018 task 9), taxonomy induction (Semeval-2016 task 13, SemEval 2015 task 17), and most notably for taxonomy enrichment (SemEval-2016 task 14) were organized for English and other western European languages in the past. However, this is the first evaluation campaign of this kind for Russian and any Slavic language. Moreover, the task has a more realistic setting as compared to the SemEval-2016 task 14 taxonomy enrichment task

[1] http://www.dialog-21.ru/evaluation/
[2] http://www.dialog-21.ru/evaluation/2020/disambiguation/taxonomia/
[3] **https://codalab.lisn.upsaclay.fr/competitions/539**
[4] https://github.com/dialogue-evaluation/taxonomy-enrichment

as the participants are not given the definitions of words but only new unseen words in context.

More concretely, the goal of this task is the following: Given words that are not yet included in the taxonomy, we need to associate each word with the appropriate hypernyms from an existing taxonomy. For example, given the input word **“утка”** *(duck)* we expect you to provide a list of its most probable 10 candidate hypernym synsets the word could be attached to, e.g. **“animal”**, **“bird”**, and so on. Here a word may refer to one, two or more “ancestors” (hypernym synsets) at the same time.

3 Evaluation metrics

We expect from participants a ranked list of 10 possible candidates for each new word in the test set. We will evaluate the systems using the Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) scores. MAP score pays attention to the whole range of possible hypernyms, whereas MRR looks at how close to the top of the list a first correct prediction is. In addition to that, the F1 score will be computed to evaluate the performance of the top 1 prediction of the methods. MAP will be the official metric to rank the submissions.

$$MAP = \frac{1}{N}\sum_{i=1}^{N} AP_i; AP_i = \frac{1}{M}\sum_{i}^{n} prec_i \times I[y_i = 1],$$

In order to be less restrictive during the evaluation, we consider as correct answers not only immediate hypernyms of new words, but also hypernyms of these hypernyms. Therefore, if a system predicted a hypernym of a correct hypernym, this will also be considered a match.

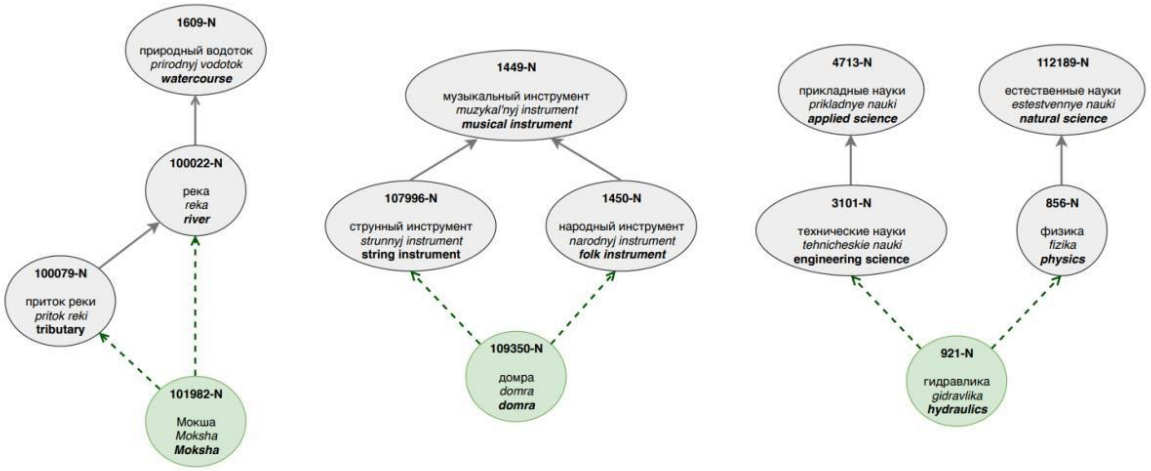
However, the specificity of the ruWordNet taxonomy and our assumption about second-order hypernyms may result in confusion in the evaluation process. Let us consider the following examples

One hypernym may be a “parent” of another hypernym (synset “Moksha” has two parents “tributary” and “river”, whereas “river” is the hypernym for “tributary”). While computing MAP score, it may not be clear which hypernym group gains the score: the one with “river” synset as immediate hypernym or “river” as second-order hypernym.

Hypernyms may share common parents: “string instrument” and “folk instrument” have the hypernym “musical instrument” in common. In this case if “musical instrument” appears in the candidate list, MAP score will also be confused. In order to avoid this hypernym ambiguity, we split immediate and second-order hypernyms into separate groups.

Each group corresponds to the connectivity component in the subgraph reconstructed from these hypernyms. We see that the first and the second subgraphs possess only one connectivity component in comparison to the last subfigure, where immediate hypernyms form different hypernym groups.

Therefore, the list of possible candidates of a given word should contain hypernyms (at least one) from each hypernym group.



4 Method

Your task is to solve the taxonomy enrichment task using a method of your choice. You can read reports of the organizers and/or reports of participants to get some inspiration. It is OK to simply reproduce some method from one of the winning participants, however note that updated models live ELECTRA appears almost every month and you may get a better quality with them.

The simple schema which could work is to apply standard word embeddings (fastText), as they do not require any additional data or training for the out-of-vocabulary words and incorporate subword tokens.

Baseline comprises the following steps:

1. Compute embeddings of all synsets in RuWordNet by averaging embeddings of all words from senses belonging to a synset.
2. Get embeddings for orphans. For multi-word orphans the embeddings are computed by averaging vectors for all words comprising an orphan.

3. For each orphan compute the top k = 10 closest synsets of the same part of speech as the orphan using the cosine similarity measure.
4. Extract hypernyms for each of these closest synsets from the previous step. Take the first n = 10 results (as each synset may have several hypernyms).

5 Results

You are supposed to test your approach on both train, validation and test sets. The best models to be submitted to the codalab platform so they are visible in the leaderboard.[5] You are normally supposed to try improving at least over the baseline approach which is available in the repository.

6. Misc: what if you do not understand Russian?

Given that both tasks are dealing with data in Russian, some basic knowledge of Russian will be useful. The use of online translators will be normally sufficient. However, if you do not possess sufficient knowledge of Russian and feel that it may be not comfortable for you, you can alternatively work on the SemEval tasks dealing with the English text data: SemEval-2010 for word sense induction[6] and SemEval-2015 for taxonomy induction[7]. The data and the evaluation scripts are provided on their websites. In case you choose one of these tasks, you do not have to take part in the competition itself (via codalab), but you need to obtain the results on the test data with an evaluation script by yourself and provide the results in your report.

7. Evaluation criteria

Technical report		**Code**		**Results**		**Total**	**Penalty for late submission**
Methodology	Discussion of results	Readability	Reproducibility	Improved over the baseline	top-1 - 10 points top-20% - 5 points	**100% + bonus**	
5	5	5	5	5 or 10	0 or 5 or 10	**25** + (5 or 10)	1 day = 1 point

* To get 100% for this task you need to achieve 25 points, but you can get additional 5 points if your method is in the top 20% (among all enrolled students) in the Codalab leaderboard and additional 10 points if your method is the top-1 in the Codalab

[5] **https://codalab.lisn.upsaclay.fr/competitions/539**

[6] https://www.cs.york.ac.uk/semeval2010_WSI/taskdescription.html

[7] http://alt.qcri.org/semeval2015/task17/

leaderboard. These credits will be counted proportionally towards the final grade in the course.

For the both tasks, you are expected to provide:

1. **Technical report (10 points total).** Write a report in the provided Ipynb template[8] describing the method used in your solution. The report must have two parts:
    a. **Methodology (5 points)**: the main of your report with description of all methods that you tried and, most importantly, that worked the best for you. Here you can include some tricks of your preprocessing, description of the models and motivation of their usage, the description of the training process details (train-test split, cross-validation, etc.). So, everything valuable that will help us to understand the scope of your work and reproduce your pipeline.
    b. **Discussion of results (5 points)**: here we want to see the final table with comparison of the baseline and all tried approaches you decided to report. Even if some method did not bring you to the top of the leaderboard, you should nevertheless indicate this result and a discussion, why, in your opinion, some approach worked and another failed. Interesting findings in the discussion will be a plus.

2. **Code (10 points total)**. Develop yourself a solution of the task and provide a reproducible code in the provided template. Make sure that your code:
    a. Is using Python 3;
    b. Contains code for installation of all dependencies;
    c. Contains code for downloading of all the datasets used;
    d. Contains the code for reproducing your results (in other words, if a tester downloads your notebook she should be able to run cell-by-cell the code and obtain your experimental results).

    As a result, you code will be graded according to these criteria:
    a. **Readability (5 points)**: your code should be well-structured preferably with indicated parts of your approach (Preprocessing, Model training, Evaluation, etc.).
    b. **Reproducibility (5 points)**: your code should be reproduced without any mistakes with “Run all” mode (obtaining experimental part).

- **Results (5 points + extra 5 or 10 points):** Push the (best) solutions which you developed to the **CodaLab** platform so that they appear in the respective public

[8] https://colab.research.google.com/drive/1IyvayaR7KS9JyofdQ14b8J3kwJ8QCLrj?usp=sharing

leaderboard. The name of your user / submission should be present in the report for verification.

- ○ You will get **5 points for outperforming the baseline**; then **additional 5 points for being in top 20%** at the public leaderboard on the private dataset OR **additional 10 points for being top 1** at the private leaderboard.

Additional notes:

Please follow these rules:

1. You should work on the model on your own and submit your own solution.
2. Use of data:
   a. The only labeled data you are allowed to use is the data provided in the CodaLab competition.
   b. You can use any unlabelled datasets you need, provided that they are open. You should specify the additional data you use in the model description (in CodaLab) and in the report.
3. In order to get the full mark you should submit your solution before the deadline. The solutions submitted after the deadline will also be checked, but only if they significantly outperform the baseline.