Assignment 1: Word Sense Induction or Morphological Analysis

1. Introduction

- Please read the document very carefully. If you have questions ask in the telegram group.
- Select ONE task: either the task from Section 2.1 (Word sense induction, WSI) OR the task from Section 2.2 (Morphological analysis).
- Use this template to complete your assignment and upload it to Canvas: https://colab.research.google.com/drive/1IyvayaR7KS9JyofdQ14b8J3kwJ8QCLrj?usp=sharing
- For WSI task (Section 2.1) not forget to submit your results to Codalab (both Practice and Test tracks): https://competitions.codalab.org/competitions/36019
- For Morphological analysis (Section 2.2) you do not need to submit to Codalab.

2. Tasks Description

In the context of this assignment, you will solve one of the tasks at your choice: word sense induction or morphological analysis. All tasks were formulated earlier in the context of the "Dialogue Evaluation"[1] campaign (RUSSE-2018: Word Sense Induction[2] or MorphoRuEval-2017: an Evaluation Track for the Automatic Morphological Analysis Methods for Russian) and or the SIGMORPHON[3] Shared Task campaign.

You are free to select one of these tasks within the framework of this assignment.

2.1 Word Sense Induction

The goal of this assignment is to use methods of distributional semantics and word embeddings to solve word sense induction. The task require knowledge of lexical semantic i.e. meaning of individual words and terms in context, e.g. the meaning of the

[1] http://www.dialog-21.ru/evaluation/
[2] http://www.dialog-21.ru/evaluation/2018/disambiguation/
[3] https://sigmorphon.github.io/

word “python” in the context “I will write my assignment using Python” is different from the meaning of the same word in the context “Pythons, are a family of nonvenomous *snakes* found in Africa, Asia, and Australia”.

Below we present information about the task. You can obtain additional information at the web site of the competition and the report of the organisers[4]. You can participate in this task by taking part in the “Word Sense Induction and Disambiguation for the Russian Language“ competition.

2.1.1 Description of the task

Word Sense Induction (WSI) is the process of automatic identification of the word senses. While evaluation of various sense induction and disambiguation approaches was performed in the past for the Western European languages, e.g., English, French, and German, no systematic evaluation of WSI for Slavic languages is available at the moment.

TLDR of the task: You are given a word, e.g. bank and a bunch of text fragments (aka “contexts”) where this word occurs, e.g. bank is a financial institution that accepts deposits and river bank is a slope beside a body of water. You need to cluster these contexts in the (unknown in advance) number of clusters which correspond to various senses of the word. In this example, you want to have two groups with the contexts of the company and the area senses of the word bank.

Namely, your goal is to fill the column **predict_sense_id** in each file with an integer identifier of a word sense which corresponds to the given context. You can assign sense identifiers from ANY sense inventory to the contexts. They should not match certain gold standard inventory (we do not provide any test sense inventory). The contexts (sentences) which share the same meaning should have the same predict_sense_id. The context will use different meanings of the target word, e.g. bank (area) vs bank (company) should have different sense identifiers.

The list of the datasets that are used for this task:

1. **wiki-wiki** located in *data/main/wiki-wiki*: This dataset contains contexts from Wikipedia articles. The senses of this dataset correspond to a subset of Wikipedia articles.

[4] https://russe.nlpub.org/2018/wsi/

2. **bts-rnc** located in *data/main/bts-rcn*: This dataset contains contexts from the Russian National Corpus (RNC). The senses of this dataset correspond to the senses of the Gramota.ru online dictionary (and are equivalent to the senses of the Bolshoi Tolkovii Slovar, BTS).

3. **active-dict** located in *data/main/active-dict*: The senses of this dataset correspond to the senses of the Active Dictionary of the Russian Language a.k.a. the ‘Dictionary of Apresyan’. Contexts are extracted from examples and illustrations sections from the same dictionary.

In the end, you need to apply your models to the three mentioned above datasets and generate three *test.csv* files corresponding to your solutions of these datasets. You can use different models to solve different datasets.

More details you can find at the GitHub repository[5]. There you can also find datasets (test sets for submissions are located in corresponding folders *data/main*) and examples to obtain some baseline solutions.

2.1.2 Evaluation metrics

Each training data contains a target word (the word column) and a context that represents the word (the context column). The gold_sense_id contains the correct sense identifier. For instance, take the first few examples from the **wiki-wiki** dataset:

The following context of the target word “замок” has id “1”:

“замок владимира мономаха в любече . многочисленные укрепленные монастыри также не являлись замками как таковыми — это были крепости…”

and all the contexts of the word “замок” which refer to the same “building” sense also have the sense id “1”. On the other hand, the other “lock” sense of this word is represented with the sense id “2”, e.g.:

“изобретатель поставил в тыльный конец ригеля круглую пластину , которая препятствовала передвижению засова ключом , пока пластина (вращаемая часовым механизмом) не становилась…”

Your goal is to **design a system which takes as an input a pair of (word, context) and outputs the sense identifier**, e.g. “1” or “2”. This is important to note that it does not matter which sense identifiers you use (numbers in the “gold_sense_id” and

[5] https://nlpub.github.io/russe-wsi-kit/

“predict_sense_id” columns)! It is not needed that they match sense identifiers of the gold standard! For instance, if in the “gold_sense_id” column you use identifiers {a,b,c} and in the “predict_sense_id” you use identifiers {1,2,3}, but the labelling of the data match so that each context labeled with “1” is always labeled with “a”, each context labeled with “2” is always labeled with “b”, etc. you will get the top score. Matching of the gold and predict sense inventories is not a requirement as we use clustering based evaluation, namely we rely on the Adjusted Rand Index. Therefore, your cluster sense labels should not necessarily correspond to the labels from the gold standard.

Thus, the successful submissions will group all contexts referring to the same word sense (by assigning the same predict_sense_id). To achieve this goal, you can use models which induce sense inventory from a large corpus of all words in the corpus, e.g. Adagram or try to cluster directly the contexts of one word, e.g. using the k-Means algorithm. Besides, you can use an existing sense inventory from a dictionary, e.g. RuWordNet, to build your modes (which again do not match exactly the gold dataset, but this is not a problem).

During the training phase of the shared task, you are supposed to develop your models, testing them on the available datasets. You will be supposed to apply the developed models to the test data, once they will be made available.

2.1.3 Method

Your task is to solve the word sense induction task using a method of your choice. You can read reports of the organisers 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 back in 2018 models like BERT did not exist, so you possibly can get much better performance with more recent models.

The simple schema which could work is to build vector representation of contexts in some way (e.g. using pre-trained models like word2vec or BERT) and then perform clustering of these contexts using such algorithms like Agglomerative Clustering or Affinity Propagation. More details can be also found in this lecture (starting from slide 36).[6]

[6] http://panchenko.me/slides/graphs-lecture.pdf

2.1.4 Results

You are supposed to test your approach on three test collections and report results on both train, validation and test sets. The best models should be submitted to the codalab platform (one per each dataset) so they are visible in both Practice and Test leaderboards[7]:

- wiki-wiki dataset
- bts-rnc dataset
- active-dict dataset

2.2 Shared Tasks on Morphological Analysis

Select one of the tasks:

- MorphoRuEval-2017: an Evaluation Track for the Automatic Morphological Analysis Methods for Russian: http://www.dialog-21.ru/en/evaluation/2017/morphology/
- SIGMORPHON 2016 Shared Task
- CoNLL–SIGMORPHON 2017 Shared Task
- CoNLL–SIGMORPHON 2018 Shared Task
- SIGMORPHON 2019 Shared Task
- SIGMORPHON 2020 Shared Task

You can choose any of these tasks.

You are expected to provide a detailed report in the same format[8]. You may want to reproduce one of the participated systems from scratch as the part of this task: http://www.dialog-21.ru/media/3966/arefyevnvermolaevpa.pdf

- If you solve this task, **do not submit** it to Codalab!
- To encourage you to do one of these tasks, you will get **10 points instead of 5 for outperforming the shared task baselines**. The BONUS part for these tasks is 5 points. It is assigned to you by TAs according to the originality / quality of your results (it is not required in this case to be in top-1 of this competition overall to get a bonus).
- If you use in your solution code from somebody, you will be required to indicate the source. In principle, your solution will be graded higher if you implement

[7] https://competitions.codalab.org/competitions/36019
[8] https://colab.research.google.com/drive/1IyvayaR7KS9JyofdQ14b8J3kwJ8QCLrj?usp=sharing

everything from scratch (re-implement it) without using publicly available solutions.

- Write a detailed report about your experiments and results.

3. Evaluation criteria

Technical report		**Code**		**Results**		**Total** **100% + bonus**	**Penalty for late submission**
Methodology	Discussion of results	Readability	Reproducibility	Improved over the baseline	top-1 - 10 points top-20% - 5 points		
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 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[9] 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

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

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 5-10 points):** Push the (best) solutions which you developed to the **CodaLab** platform so that they appear in the respective public leaderboard. The name of your user / submission should be present in the report for verification.
    - **For the WSI task**, firstly, you will get **5 points for outperforming the baseline**; then additional 5-10 points for being in top 20% at the public leaderboard on the private dataset.
    - **For Morphological Analysis** you will get **10 points for outperforming the baseline**; 5 points can be assigned to you for the originality and quality of your results.