Assignment 1: Word Sense Induction

1. Introduction

- Please read the document very carefully. If you have questions ask in the telegram group of the course (link can be found in Canvas).

- Use this template to complete your assignment and upload it to Canvas: https://colab.research.google.com/drive/1ttPT6X4K0ovgbzmNjlcEiprkj1LaBuF2#scrollTo=8a46ab45-d215-41af-b910-63ff4a215a07

- Submit your results to Codalab (both Practice and Test tracks): https://codalab.lisn.upsaclay.fr/competitions/8322

    - **Important**: valid submissions of this assignments must indicate the following team name “Intro to NLP - 2023”.

2. Tasks Description

In the context of this assignment, you will solve word sense induction task as formulated in the context of the “Dialogue Evaluation”[1] campaign (RUSSE-2018: Word Sense Induction[2].

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 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”.

[1] http://www.dialog-21.ru/evaluation/
[2] http://www.dialog-21.ru/evaluation/2018/disambiguation/

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[3]. 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. In this task, you are given a word, e.g. bank and a set 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.

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.

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

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[4]. 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 “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.

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

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 ideas about methods can be obtained from:

- Reading reports of the participants of the original shared task.[5]
- While studying literature on the “word sense induction”, “word in context / wic” task at ACL Anthology web site. Just look for papers and get ideas usable here. [6]

The basic approach is to pick some pre-trained encoder or word embedding model, represent context with it and then perform clustering of the contexts. You may try other i.e. supervised setups too.

[5] https://www.dialog-21.ru/evaluation/2018/disambiguation/
[6] https://aclanthology.org/

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.