Politecnico di Torino
Dipartimento di Automatica e Informatica
Deep NLP (01VIXSM)
Written Exam

February 11, 2022

Name: ____

Surname: ____

Student ID: ____

Exam rules:

- The present exam consists of 6 pages (including this cover page) and 7 questions overall. Any inconsistencies/printing errors in the written exam content must be reported to the teacher *at the beginning* of the exam.
- Exam duration: *60 minutes*.
- Withdraw is allowed only *at the end* of the exam.
- The exam is *closed-book*. Electronic devices, mobile phones, smart watches, and extra papers (even blank papers) are *not allowed*.
- Closed-ended questions: cross the right answer (just one) at pag. 2. Wrong or misszing answers to closed-ended questions will receive *no penalty*.
- Open questions: write your answers below the text of the question. If you need more space please use the last page (i.e., pag. 6) and/or the back side of the paper.

Evaluation grid

Question:	1	2	3	4	5	6	7	Total
Points:	1	1	1	1	5	5	6	20
Score:								

1. (1 point) Which of the following statements holds *true* for the BERT pre-training phase?
   - It uses a context window to define positive examples.
   - Self-supervised training with masked language modeling and next sentence prediction is used in the training phase.
   - The model is trained using Next Word Prediction task.
   - It is trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.
   - None of the above.
2. (1 point) The Negative Sampling
   - Can be used for bootstrapping supervised classification models.
   - Can be used to avoid data overfitting.
   - Can be used to find negative words in sentiment analysis.
   - Can be used to select negative examples while training a Word2Vec model.
   - None of the above.
3. (1 point) Considering the one hot encoding representation
   - Each textual unit is represented by a dense vector consisting of real-valued elements.
   - Each textual unit is represented by a sparse vector consisting of boolean elements.
   - Each textual unit is weighted by its number of occurrences in the input corpus.
   - It varies according to the size of the context window.
   - None of the above.
4. (1 point) Considering the HITS algorithm, during the HUB update step
   - Each node relevance score is normalized by the number N of nodes.
   - Each link relevance score is normalized by the number N of nodes.
   - For each node, the hub score is the sum of the authority scores of each node that it points to.
   - For each node, the hub score is the sum of the hub scores of each node that points to it.
   - None of the above.

5. (5 points) Explain the **Latent Dirichlet Allocation**:

    1. Elaborate on the steps required to generate an LDA model.
    2. Describe the Author-Topic Model (ATM) and its similarities/differences with LDA.
    3. Enumerate <u>at least</u> two practical examples of application of ATM.

Draft solution 5.1:

- Generative topic model
- Each word in a document is assumed to be generated either by sampling a topic from a document-specific distribution over topics and by sampling a word from the distribution over words that characterizes that topic.
- For each document in the corpus and for each term, a topic is chosen accordingly to the document-topic distribution.
- Words are extracted from the input vocabulary V by taking into account the terms probabilities for each given topic in the document mixture.

Draft solution 5.2:

It is a Generative model for documents and extends the Latent Dirichlet Allocation to include authorship information.

1. Each author is associated with a multinomial distribution over topics
2. Each topic is associated with a multinomial distribution over words
3. A document with multiple authors is modeled as a distribution over topics that is a mixture of the distributions associated with the authors

Draft solution 5.3:

The author-topic model can be used for:

- Who is the most authoritative author on a given topic?
- What are the topic covered by a given author?
- What is the most authoritative paper of an author?

6. (5 points) Elaborate on the **Recommendation** task.

    1. Formulate the task and clarify the main goals.
    2. Illustrate at least two business scenarios of usage for a recommender system.
    3. Compare *content-based* and *collaborative filtering* systems by highlighting pros and cons of each of the above-mentioned strategies.

Draft solution 6.1:

1. Let U be a set of users, I be a set of recommendable items, R an ordered set of ratings, the task is find $F(\cdot)$: $U \times I \rightarrow R$. The goal is to generate user-specific item rankings.

Draft solution 6.2:

1. NetFlix users receive movie recommendations based on their previous interactions with the platform.
2. Travelers of a tourism agency receive hotel recommendations based on the census data (e.g., age, gender, job, salary, etc.)

Draft solution 6.3:

1. Collaborative filtering: recommend to a given user those items that were selected by similar users. Pros: no need for content-level explorations (more efficient). Cons: popularity bias, cold start.
2. Content-based approaches: recommend items that are most similar to those previously selected by the same user. Pros: solves the cold start and the first rater problems. Cons: filter bubble. Need for content-level explorations (more computationally intensive).

7. (6 points) The *Politecnico di Torino* aims at detecting hate speech in student-written Twitter posts. PoliTO NLP engineers have at their disposal

   - a list of Twitter users (i.e., the students' identifiers).
   - the official Twitter APIs that allow them to crawl all the tweets posted by a user with a given identifier within a delimited time period.
   - Any *opensource* data collection, NLP and ML libraries available on the Web.
   - a budget of 200 humanly-generated annotations.

   The final goal is to detect the tweets containing hate speech and the students that most frequently wrote hateful Twitter posts.

   Notice that it is *not possible* to make any arbitrary assumption on the language of the Twitter posts.

   1. Design a complete NLP pipeline for accomplishing task. Engineers should be able to report the performance of the system leveraging *only* the resources at their disposal.
   2. How can engineers increase the quality of the proposed system by leveraging on additional resources (*other than* those already at their disposal)?
   3. Provide an example of the evaluation methodology for step 1.

   Draft Solution 7.1:

   Key points:

   - Detect tweet language and apply MT model if required (all tweets in the same language, e.g., English)
   - Use open-source data collections to train a sentiment analyzer (text classification model) for hate speech detection.
     - Fine-tune an encoder model (e.g., BERT, RoBERTa) or,
     - Train a Machine Learning model on top of unsupervised text representations (e.g., Word2Vec, Paragraph2Vec).
   - Performance analysis on the provided annotations (200 humanly-generated annotations).

   Draft Solution 7.2:

   - Ask for additional manual annotations to fine-tune an encoder-based model on specific hate-speech detection task.
   - Collect lexicon of hate-related words to weigh sentence tokens.

   Draft Solution 7.3:

   - Use the annotations provided by humans to assess the performance of the classification system (no additional annotations).
   - Compute classification metrics: accuracy, precision, recall, f1-score and confusion matrix (add definition).

This page is intentionally left blank to accommodate work that wouldn't fit elsewhere and/or scratch work.