Exercises

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino

Politecnico
di Torino

Exercise nr. 1

An hotel chain wants to analyze the 10,000 reviews submitted by the hotel guests. Reviews are submitted through a HTML form. Notice that the review language is a priori unknown.

- Enumerate the steps required to build a sentiment analyzer based on SVM classifier applied to a tf-idf-based incidence matrix.
- Motivate the adoption of each step.

A literature review and classification of recommender systems research. Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. https://doi.org/10.1016/j.eswa.2012.02.038

Exercise nr. 1: draft solution

- Filter out non-textual content or special characters
- Detect the source language and perform Machine Translation separately for each text snippet (if need be)
- Apply word tokenization (using language-dependent separators)
- Retrieve a (language-dependent) stopword list and remove them
- Apply lemmatization/stemming
- Compute the tf-idf matrix (formula)
- Build a co-occurrence-based relational data representations
    - One row per review, one column per word plus the class label
- Select the most relevant features using supervised feature selection techniques
- Split labeled reviews into train (e.g., 7000), validation (e.g., 1000), and test sets (e.g., 2000)
- Train a SVM classifier on training data
    - Validate the model settings on the validation set
    - Test the model on the test set

Exercise nr. 2

An hotel chain wants to analyze the 10,000 reviews submitted by the hotel guests. Reviews are submitted through an HTML form and can be written in different languages.

- Enumerate the steps needed to train a sentiment analyzer using BERT
- Motivate the adoption of each step

A literature review and classification of recommender systems research. Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. https://doi.org/10.1016/j.eswa.2012.02.038

Exercise nr. 2: draft solution

- Detect the language
- Filter out non-textual content or special characters
- Apply word tokenization (using language-dependent token separators)
- Truncate the sentences that contain more than 512 tokens
- Split the labeled reviews into train (e.g., 7000), validation (e.g., 1000), and test sets (e.g., 2000)
- Encode sentences using Multilingual BERT

  https://huggingface.co/bert-base-multilingual-cased
  https://github.com/google-research/bert/blob/master/multilingual.md

- Fine-tune BERT for sentiment analysis
- Test the model on the test set

Exercise nr. 3

A journal editor wants to process the section-level content of English-written scientific papers submitted for blind review in order to automatically categorize sections as introductory, methodological, empirical, *or* conclusive. *Papers are submitted in PDF format. The number of available papers is 1000. Each paper contains on average 5 sections.*

- What are the text preprocessing steps needed to train text classifier based on SVM classifier applied to a W2V-based pretrained word representation?
- Motivate the adoption of the each step

A literature review and classification of recommender systems research. Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. https://doi.org/10.1016/j.eswa.2012.02.038

Exercise nr. 3: draft solution

- Convert the PDF file into text
- Remove special characters, conversion errors, and non-textual content
- Performing sectioning
  Each document is a different section title
- Perform word tokenization
- Look for each word into the pretrained Word2Vec model dictionary
  Missing words are neglected
- Average word-level features over documents
  Aggregate word-level encoding into document-level ones
- Build a labeled relational dataset
  Each row is a document, each column is a latent feature in the high-dimensional vector representation
- Apply feature selection steps (if necessary)
- Split the labeled data into train, validation, and test sets
- Train a SVM classifier on training data
- Validate the model settings on the validation set
- Test the model on the test set

Exercise nr. 4

A publisher wants to recommend additional books to users who did early access the books available in the online store. The available data consist of a large set of book-related information, i.e., book title, abstract, and keywords. User information is not available.

- What kind of recommender system can be designed for?
- Formalize the problem.
- Discuss the pros and cons of each candidate solutions and the main blocks of the preferred one.

Exercise 4: draft solution

Content-based item-to-item recommender

Problem statement

Given a book catalog C, the task is to infer a similarity function $F : B \times B \rightarrow R$, that scores the similarity between any pair of books b_1, $b_2 \in B$.

Approaches

- Word-level text similarities to compare book titles and abstracts
  - W2V -> can be specialized into the target domain
  - FastText -> handles zero-frequency terms
  - GloVe -> more suitable for short text snippets
- Contextual similarities
  - RecoBERT-like approach
    - BERT is suitable for comparing title and abstract
    - GloVe/ELMo are suitable for comparing keywords

Data model

Each book is a triple <title, abstract, keyword>
Title and abstract can be processed together by the BERT encoder
BERT and ELMo encodings can be concatenated and jointly provided as input to the RecoBERT architecture
REcoBERT computes the similarities between positive-negative book pairs

Exercise nr. 5

Design a NLP pipeline aimed at summarizing the description of each course in a 100-word abstract to be visualized in the Polito Webpage. At this stage, let us assume that no human-generated annotations are provided. Enumerate the main steps, the names of the models and algorithms used, and any further settings/relevant assumptions made in the design process.

Some volunteers provided manually generated summaries of 1000 course descriptions. How can we exploit the annotated data in order to improve the quality of the generated summaries?

What are the procedures and metrics used to evaluate the performance of the summarization pipeline at Step 2?

Exercise nr. 5: draft solution

- Language identification
- Separately for each document, apply an unsupervised summarization method to select the most relevant sentences (e.g., TextRank)
  - Explain the selected method
- Exploit the annotated data to finetune a pretrained model (e.g., BERTSUM)
  - motivate the choice
- Rouge: explain the selected metric.
- Which Rouge score? Recall (Fixed summary length)
- Describe how to compare the generated summaries with the golden summaries.

Exercise nr. 6

Given (1) a collection of 2000 English-written scientific papers (for the sake of simplicity, neglect the non-textual content), (2) the list of paper authors, and (3) the co-authorship information (i.e., who are the co-authors of each paper), we would like to design a "co-author recommender system". Specifically, given an author we want to find the top-K best candidate co-authors.

In a nutshell, "I am an author. Among the persons that are currently available in the paper authors' list, who are my best candidate collaborators, excluding those that have already been co-author of mine?".

1. Design a complete NLP pipeline for accomplishing the task. For each pipeline step clarify the goal, the algorithms used, and the main settings.
2. Describe the metrics used to evaluate the system performance and explain their meaning.

Exercise nr. 6.1: draft solution

The model aims at proposing a solution for the task of recommending co-authors to researchers. Hereafter a draft of the pipeline:

- Collect, for each user, the document that he/she has co-authored.
- Generate a graph to represent both users and publications. Each author as well as each publication is a node in the graph.
- Create edges e connecting authors with each publication he/she co-authored (collaborative filtering)
- Create edges e between publications according to their semantic similarity (content-based)
  - Use SBERT model to encode the abstract of each publication.
    - a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings (https://arxiv.org/abs/1908.10084)
  - Compute the similarity between two papers using the cosine similarity between their embeddings.
- For each author a, provide a list of possible co-authors that maximize the sum of the connections between the two: $s_{a \to i} = \sum_w e^w_{a \to i}$
- Remove from the list all the authors that have at least one publication in common.
- The list can be ordered by decreasing $s_{a \to i}$

Exercise nr. 6.2: draft solution

We can generate a train-test split of the data collection and use some of the papers to evaluate the model. Specifically, each paper left for testing is used as ground-truth for evaluation.

Metrics

- Mean Reciprocal Rank (MRR): the average of the reciprocal rank.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

- Precison or Mean Average Precision (MAP): percentage of recommended relevant items.

Exercise nr. 7

Given a financial news you want to detect the relations between different text spans. Specifically, the goal is to "identify and extract", within a sentence or a longer text block contained in the news content, the causal elements and the consequential ones. Assume that they consist of distinct cause-effect pairs.

Example:

Original news content: *Zhao found himself 60 million yuan indebted after losing 9,000 BTC in a single day (February 10, 2014)*

Extracted cause: *losing 9,000 BTC in a single day (February 10, 2014)*

Extracted effect: *Zhao found himself 60 million yuan indebted*

Assume that a ground-truth consisting of 1000 <*content*, *cause*, *effect*> triples are given.

1. Design a complete NLP pipeline for accomplishing the task. For each pipeline step clarify the goal, the algorithms used, and the main settings.
2. Describe the metrics used to evaluate the system performance and explain their meaning.

Exercise nr. 7.1: draft solution

- Language identification and machine translation (if need be)
- Selection of a machine learning model able to classify single tokens. In this case, it could be useful to select a pre-trained encoder model (e.g., BERT)
- Text tokenization using a tokenizer selected according to the previous step.
- Training/Fine-tuning a token classification model (similar to NER).
- Examples of annotations used in the training phase:
  - B-CAUSE: begin token for the clause
  - I-CAUSE: token inside a clause (including final token)
  - B-EFFECT: begin token for the effect
  - I-EFFECT: token inside a effect (including final token)
  - OTHER: token outside clause and effect
- Running the fine-tuned model on inference model (e.g., BERT) to detect cause/effect snippets.

Exercise nr. 7.2: draft solution

The following two metrics can be used to evaluate the proposed system

- ROUGE-based scores can be used to identify the overlap between the system and reference text snippets.
- Exact match can be used to compute the fraction of cause/effect that are correctly identified.

Acknowlegdements and copyright license

- Copyright licence
  - Attribution + Noncommercial + NoDerivatives

- Acknowledgements
  - I would like to thank Dr. Moreno La Quatra, who collaborated to the writing and revision of the teaching content
- Affiliation
  - The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
    - https://dbdmg.polito.it
    - https://smartdata.polito.it

Thank you!

