2025 Information Retrieval and Extraction

HW 1

Task introduction

- Measure document relevance to a query
  - Implement **vector model** and **BM25** using only numpy to compare query and code snippet relevance
  - Apply Dense Retrieval with a pre-trained model, and compare it with a fine-tuned version using **train_queries.csv**
  - You must implement TF-IDF and BM25 by yourself
    (Don‘t use **BM25Okapi**, **cosine_similarity**, **TfidfVectorizer**, etc.)
- Requirement
  - Upload your submission to Kaggle
  - Submit a report and your source code to E3
- Deadine is 11/11 (Tue.) 23:59, no late submission

Dataset

- code_snippets.csv
  - Code snippets and their corresponding Code IDs.
- train_queries.csv
  - Contains queries and corresponding code snippets.
- test_queries.csv
  - Contains queries that need to be used for prediction.
- sample_submission.csv
  - a sample submission file in the correct format.

Training Data

	code	query
1	ped_func) ensure_callback_server_started(gw) return self	batch_id): ... batch_df.collect() ... >>> writer = sdf.writeStream.foreach(func)
2	_vcf(fnames, batch_id, caller, data)) return caller_names, vrn_files	Retrieve variant calls for all samples, merging batched samples into single VCF.
3	ype.data_type) data_type.data_type = resolved return resolved	(DataType): The target DataType/Alias to resolve. Return: DataType: The resolved type.
4	self._sout.write() else: self._sout.write(% taskid)	Show stack frames for a task
5	numpy") except Exception as e: print_error(e) return True	Try to import the aeneas package and return ``True`` if that fails.
6	negate=negate, preserve_case=preserve_case) return ii_node	emRestoreItem/OriginalFileName :return: A IndicatorItem represented as an Element node
7	return ret curr_policy = bucket[].get(, {}).get(, {}) return ret	Action: s3:GetObject Resource: arn:aws:s3:::the_bucket_for_my_distribution/*
8	) self.__ipv6_phy_intf_cmds = t if hasattr(self,): self._set()	g to populate this variable should do so via calling thisObj._set_ipv6_phy_intf_cmds() directly.
9	=) logging.Formatter.converter = time.gmtime return formatter	formatter used in our syslog :param request: a request object :returns: logging.Formatter
10	else: return os.environ.get(self.name, self.default_value)	Resolve given variable
11	entDecrement()]) print >>outFile, table.get_string().encode("utf-8")	Given an instance of TemporalMemory, print out the relevant parameters
12	_name, t.DataError(.format(name)), (confirm_name,) return check_	. Checks if data['name'] equals data['confirm_name'] and both are valid against `trafaret`.
13	n Structure(b"F", seconds, nanoseconds, tz.utcoffset(value).seconds)	Dehydrator for `datetime` values. :param value: :type value: datetime :return:
14	observed.add(name.get_molecule()) return negative_filter|observed	Go through a stream and print out anything not in observed set
15	content_lines[end_line:]) self.content = .join(new_content_lines)	Inject string $b16_scheme into self.content.
16	_grr.VFSGRRClient.SchemaCls.PING) >= oldest_time: yield fd	Yield client urns.
17	result: output_writer.WriteLine() output_writer.WriteLine()	de (SourceScanNode): the locked scan node. output_writer (StdoutWriter): the output writer.
18	:width(libc, ucs) except AssertionError as err: print(err)	local wcwidth.wcwidth() function; when they differ, report a detailed AssertionError to stdout.
19	pass return submodules	lule: Module object from which to import sub-modules. :return: Dict with name-module pairs.
20	:=shortest) res = rand.data(length, self.charset) return res	her or not the shortest reference-chain (most minimal) version of the field should be generated.
21	model, reaction, , flux_coefficient_cutoff) return results	Where Status is the results from assess_precursors and assess_products, respectively.
22	url = self.api_url + + self.api_key + + archive_id + return url	this method returns the url to set the archive layout

Testing Data

	query_id	query
1	1	l bgp open message to peer and initialize related attributes.
2	2	:return: User object with `permanently_deleted` status
3	3	Start a new paragraph.
4	4	ɔ to the user to input a sensible sampling distribution!
5	5	verride this to deal with different types of object from Page.
6	6	nread will terminate if it sees a sentinel object in the queue.
7	7	ı path. :param path: Path to tasks. :return: None.
8	8	Return enforced ascii string éko=>ko
9	9	<https://bugzilla.redhat.com/show_bug.cgi?id=1235377>`_.
10	10	insert object before index
11	11	ɔming POST as a GET to work around URI length limitations
12	12	ns configuration. :param email: The email address.
13	13	turn: Object name list :rtype: tango.DevVarStringArray
14	14	ıg file is used. Returns: value from configuration file
15	15	das dataframe with excel data :rtype: pandas.DataFrame
16	16	InvalidSlot: If ``slot`` doesn't accept keyword arguments.
17	17	arameters: :return: :rtype: SnmpContextManager
18	18	in the invoice. Can be < 0 if the invoice was overpaid.
19	19	Get an instance of Api Neighbor services facade.

Requirements & Scoring Metrics

- Please implement both **TF-IDF** and **BM25** as the scoring methods for Sparse Retrieval, and additionally apply a **Dense Retrieval approach** using a pre-trained model (e.g., CodeBERT) to retrieve the most relevant code snippets.
- For each query, output the **top-10** most similar code IDs in a single line, separated by spaces.

- The final scoring result will be conducted as **Recall@10.**
  - If the ground-truth code ID appears in the top-10 retrieved results, it will be counted as 1.
  - Otherwise, it will be counted as 0.
  - The final score is the average across all queries.

Kaggle Submission

	query_id	code_id
1	1	1 1 1 1 1 1 1 1 1 1
2	2	2 2 2 2 2 2 2 2 2 2
3	3	3 3 3 3 3 3 3 3 3 3

Three numbers separated by spaces

- kaggle link
- Display team name : <student ID>
- Submission format
  - A 500*2 .csv file, first row is for the column name and the last 500 rows for your result.
  - Column name must be query_id and code_id.
- There is one simple baseline and one strong baseline. Beat them to achieve a higher score.

#	Team	Members	Score	Entries	Last	Join
	Strong Baseline		0.72000			
	Simple Baseline		0.52400			

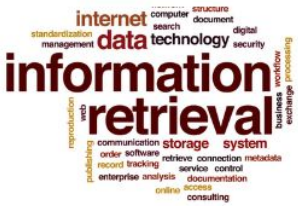
Kaggle Submission

- The scoring metric is Recall@10.
- You can submit at most 5 times each day.
- You can choose 2 of the submissions to be considered for the private leaderboard, or will otherwise default to the best public scoring submissions.
  You can only view your private leaderboard score after the competition has ended.
- Public leaderboard is calculated with 50% of the test data, and private leaderboard is calculated with other 50% of the test data, so the final standings may be different.
- Please tune your model parameters using your own validation set instead of adjusting parameters based on the public leaderboard. Otherwise, it's easy to overfit, leading to poor performance on the private leaderboard.

Change your team name

2025 Generative Information Retrieval HW1

Homework 1 for Generative Information Retrieval @ NYCU, 2025



Settings Overview Data Discussion Leaderboard Rules Team

Remember to change the team name to <student ID>, or there will be a deduction of 5 points for HW 1.

Your Team

Everyone that competes in a Competiton does so as a team - even if you're competing by yourself. Learn more.

General

TEAM NAME

Team Name

Report Submission

Answer the following 3 questions:

1. In Sparse Retrieval methods, compare the retrieval performance of TF-IDF and BM25. Which method performs better in this assignment? Analyze the possible reasons behind the difference (e.g., term frequency handling, document length normalization).

2. In Dense Retrieval methods, compare the performance of using a pre-trained model directly versus fine-tuning with training data. Which approach performs better? Explain the possible reasons for the difference.

3. In the Text-to-Code Retrieval task, compare the differences and performance between Sparse Retrieval and Dense Retrieval. Beyond these approaches, what other methods (e.g., Retrieve-and-Re-rank) could further improve retrieval performance?

Please answer the questions in detail to receive full points for each question.

Grading policy

- Kaggle (70%)
  - 30% based on the public leaderboard score and 70% based on the private leaderboard score
  - Leaderboard score consists of basic score and ranking score
    - Basic score :

      Over strong baseline : 55

      Over simple bassline : 40

      Under simple baseline : 25
    - Ranking score:

      15-(15/N)*(ranking-1), N=numbers of people in the interval
- Report (30%)
  - 10 for each quesiton

E3 Submission

Submission format:

- hw1_<student_id>.zip
  - source code: hw1_<student_id>.py or hw1_<student ID>.ipynb
  - report: hw1_<student_id>.pdf

- Submit your source code and report to E3 before **11/11(Tue.) 23:59**, **no late submissions will be accepted.**

- Failed to comply with above rules (under any circumstances) will cause a **deduction of 5 points** to your score.

If you have any question about HW 1, please feel free to contact with TA : Chun-Wei Kang

through email nick020789.cs13@nycu.edu.tw

Have Fun !