Assessing the IT Skills Gap: A Comparative Analysis of Tertiary IT Education in the Philippines and Industry Expectations

Preprint · February 2025

DOI: 10.13140/RG.2.2.22165.64488

Vince Bernard Austria
STI College

Assessing the IT Skills Gap: A Comparative Analysis of Tertiary IT Education in the Philippines and Industry Expectations

Vince Bernard Austria[1*]

[1*]Systems Technology Institute (STI), Altaraza Town Center, San Jose del Monte, 3023, Bulacan, Philippines.

Corresponding author(s). E-mail(s): vinceaustria54@gmail.com;

Abstract

The increasing demand for IT professionals in the Philippines has placed pressure on higher education institutions (HEIs) to align their curricula with industry expectations. This study examines the competency gap between the Bachelor of Science in Information Technology (BSIT) curriculum and the skills required by top IT employers and non-IT corporations. Using a mixed-methods approach, job listings from leading companies were analyzed and compared to BSIT course outlines through Natural Language Processing (NLP) techniques, including semantic similarity analysis and Term Frequency-Inverse Document Frequency (TF-IDF). Results indicate that while BSIT programs cover foundational IT competencies, they fall short in key industry-demanded skills, particularly in specialized and emerging technologies. The findings emphasize the need for curriculum enhancements to bridge the IT skill gap and ensure employability of graduates.

Keywords: IT education, skills gap, semantic analysis, industry alignment

1 Introduction

The Bachelor of Science in Information Technology (BSIT) in the Philippines was formally established in 1996, two years after the foundation of the Commission on Higher Education (CHED). Alcala (2015) This is under the passage of the Higher Education Act of 1994, which sought to create a separate and autonomous body distinct from the Department of Education, Culture and Sports (DECS), now known as the Department of Education (DepEd) Quisumbing (1998), 9th Congress of the

Philippines (1994). Prior to 1996, the BSIT Program as a distinct course did not exist, as DECS had previously limited offerings in Computing education to programs such as Computer Science and Computer Engineering.

The establishment of CHED marked a pivotal moment in the development of higher education in the Philippines, particularly enabling a more specialized responsive approach to tertiary education. Initial initiatives such as the introduction of new academic programs, that are specifically designed to address the growing demand for expertise in information technology and related fields. In addition to BSIT, CHED facilitated the development of other computer-related programs, such as the Associate in Computer Technology (ACT) and Bachelor of Science in Information Systems (BSIS) in 2006 it replaced the former program titled Bachelor of Science in Information Management (BSIM). Reflecting the department's efforts to align educational offerings to the industry standards and the evolving needs of the global technology landscape. Commission on Higher Education (2006), Commission on Higher Education (1998)

As stated by CHED's latest Revised policies, standards and guidelines for BSIT program (CMO no.25 s.2015), graduates are expected to demonstrate proficiency in key areas. These includes the ability to analyze, identify, and address complex computing problems; the application of computing knowledge to solve real world problems; the development of solutions aligned with computer systems; and the effective utilization of modern computing tools. Commission on Higher Education (2015) Such outcome is expected to reflect the program's emphasis on producing graduates that are equipped to navigate the rapidly evolving and changing technological landscape.

As specified in the Republic Act No. 7722, also known as the Higher Education Act of 1994, the Commission on Higher Education (CHED) is mandated to formulate and recommend the minimum unit requirements for academic programs 9th Congress of the Philippines (1994). This legislative framework ensures that Higher Education Institutions (HEIs) retain their autonomy in designing their own curricula, provided that it adheres to the established minimum national standards set by CHED. Given the academic freedom granted to higher education institutions in the Philippines, specific learning outcomes may vary between institutions. However, core competencies remain consistent as outlined by the Commission on Higher Education (CHED). At the same time, the IT industry continues to raise its expectations for BSIT graduates, often exceeding the minimum academic standards. As noted by industry practitioners Bringula (2016), employers place the highest value on IT skills acquired during formal education, followed closely by communication and critical thinking abilities. This shifting landscape necessitates that higher education remains agile and responsive to industry demands. Without continuous adaptation, graduates risk misalignment with workforce expectations. Thus, it's imperative for the higher education sector to be agile and dynamic to the ever changes of the industry. This is a necessity in order for BSIT graduates to be competent on their field.

The integration of computing, particularly information technology, into the Philippine business sector can be traced back to the 1960s, when IBM introduced the IBM 650 to the Bureau of Public Lands. This was followed by the installation of the IBM 1410 at Philamlife Insurance Corporation two years later Buelva (2002). Since then, computing systems and their associated technologies have permeated nearly all industries in the country, from manufacturing to telecommunications. However, despite the presence of these early computing systems, their utilization remained highly limited. The size, power consumption, and prohibitive costs of IBM mainframes restricted access to large corporations, effectively barring smaller enterprises from adopting such technologies International Business Machines (IBM) (n.d.).

A significant shift occurred in 1994 when the Philippines established its first undersea intercontinental internet connection to the United States Lallana EC (2007). This milestone marked the beginning of widespread information technology adoption. Between 2009 and 2010, internet usage in the country surged from 9% to 25% Data Commons (2024). By 2025, an estimated 89.59 million individuals had access to smartphones, with internet penetration reaching 98% Statista (2024b), Statista (2024a).

This continuous expansion of digital infrastructure has facilitated the integration of information technologies across various industries, particularly in the Business Process Outsourcing (BPO) sector. The availability of relatively low-cost skilled labor, coupled with increasing internet bandwidth, has contributed to the sector's rapid growth. In 2022, the BPO industry generated revenues of $35.9 billion, reflecting a 10.3% increase from the previous year IT & Business Process Association of the Philippines (IBPAP) (2019). This sustained growth underscores the critical role of IT in driving economic development and enhancing operational efficiency across multiple sectors. The global expansion of data processing and consumer-facing applications continues to drive advancements across multiple domains, including Web2 frameworks, cloud computing, blockchain-based financial transactions, and artificial intelligence. These developments collectively fuel the growth of the Philippine IT industry. To remain competitive, the higher education sector must proactively adapt to evolving global IT trends. A 2021 study by the Philippine Institute for Development Studies (PIDS) revealed that 29% of Filipinos lacked the technical skills required to meet industry demands Angelo (2021).

This study examines the alignment between IT curricula in Philippine higher education institutions and industry expectations, particularly for associate-level roles. By identifying competency gaps, it seeks to assess whether academic programs sufficiently equip graduates for workforce demands.

2 Methodology

This study employed a mixed-methods approach to collect, analyze and interpret quantitative data. The dataset comprises IT-related job listings from Top 10 IT Companies in the Philippines determined by Great Place to Work survey in 2024 Great Place to Work Philippines (2024) [$N = 176$] and job descriptions from the Top 10 non-IT specific Companies in the Philippines by market capitalization (*Table 1*) CompaniesMarketCap (2024) [$N = 184$]. These job listings were gathered from multiple online platforms, including LinkedIn, Indeed and company webpages. Furthermore, to ensure consistency and mitigate Large Language Model hallucination, the data was formatted using Gemini 1.5 Pro tuned multi shot text prompt model (Temp = 0, Top

P = 0) across five predefined categories: (1) Tech Stack Requirements, (2) IT-Related Qualifications & Skills, (3) Required Years of Experience, (4) Non-IT Skills required, and (5) Key Responsibilities.

Top 10 IT Companies in the Philippines Great Place to Work™ (2024)	**Top 10 Largest Companies in the Philippines by Market Capitalization (2024)**
Synchrony Global Services Philippines, Inc.	SM Investments Corporation
Cisco	SM Prime Holdings
Accenture	International Container Terminal Services
Atlassian	BDO Unibank
Via Appia Philippines, Inc.	Bank of the Philippine Islands
TaskUs	Ayala Corporation
Capital One Philippines	Ayala Land
3M GSC Philippines	Metropolitan Bank
DXC Technology Philippines	Globe Telecom, Inc.
Teleperformance	Jollibee Foods Corporation

Table 1: Top IT and Largest Companies in the Philippines (2024)

The BSIT sample curriculum course outline and dataset was derived from CHED Memorandum Order No. 25, Series of 2015 (CMO 25, s. 2015) (*Table 2*) and collected from 34 randomly selected higher education institutions in the Philippines (*Table 3*). Each individual coursework was standardized using the same Gemini 1.5 Pro multi shot model into four predefined categories: (1) Tech Stack Learned from the coursework, (2) IT-Related Qualifications & Skills Learned from the coursework, (3) Non-IT Skills Learned, and (4) Key Responsibilities Learned. The "Years of Experience" category was omitted, as students are not yet industry professionals, though the data was still collected for consistency.

Course Code	**Area Code**	**Course Title**	**Units**
CC101	CP	Introduction to Computing	3
CC102	PF	Computer Programming 1	3
CC103	PF	Computer Programming 2	3
CC104	PF	Data Structures and Algorithms	3
CC105	IM	Information Management	3
CC106	WS	Application-Development and Emerging Technologies	3
HCI101	HCI	Introduction to Human Computer Interaction	3
IAS101	IAS	Information Assurance and Security 1	3
IAS102	IAS	Information Assurance and Security 2	3
IM101	IM	Fundamentals of Database Systems	3
IPT101	IPT	Integrative Programming and Technologies 1	3

Continued on next page

Course Code	Area Code	Course Title	Units
MS101	MS	Discrete Mathematics	3
MS102	MS	Quantitative Methods (including Modeling & Simulation)	3
NET101	NET	Networking 1	3
NET102	NET	Networking 2	3
PRAC101	PRC	Practicum	3
SA101	SA	Systems Administration and Maintenance	3
SIA101	SIA	Systems Integration and Architecture 1	3
SP101	SP	Social and Professional Issues	3
CAP101	THS	Capstone Project and Research 1	3
CAP102	THS	Capstone Project and Research 2	3
Recommended Electives			
IPT102	IPT	Integrative Programming Technologies 2	3
PT101	PT	Platform Technologies	3
WS101	WS	Web Systems and Technologies	3
PF101	PF	Object-Oriented Programming	3
SIA102	SIA	Systems Integration and Architecture 2	3
HCI102	HCI	Human Computer Interaction 2	3

Table 2: BSIT Sample Curriculum Guidelines

List of Randomly Selected HEIs	
A-G	**H-Z**
ACTS Computer College	Holy Cross of Davao College
Aldersgate College	Institute of Business, Science and Medical Arts
Baliwag Polytechnic College	Isabela State University
Bulacan State University	Malilig Plains Colleges, Inc.
Cagayan State University	Manila Tytana College
Cedar College Inc	Misamis University
Central Philippines State University	Nueva Ecija University of Science and Technology
Colegio de Montalban	Nueva Vizcaya State University
Colegio De San Jose	Pamantasan ng Lungsod ng Muntinlupa
De La Salle University	Pangasinan State University
Don Mariano University	Philippine College of Technology
Dr. Aurelio Mendoza Memorial Colleges	Quezon City University
Eastern Visayas State University	Saint Anthony College
FEU Roosevelt	University of Eastern Philippines
Gordon College Olongapo City	University of Makati
	University of the Philippines Baguio
	Virgen Milagrosa University Foundation

Table 3: List of randomly selected HEIs for course outline

To assess the skill gap between Bachelor of Science in Information Technology (BSIT) curricula and industry demands, Natural Language Processing (NLP) techniques were applied using Python. Semantic similarity analysis was performed to quantify the alignment between academic curricula and job listings, generating semantic similarity scores and a relationship plot between job postings from the top 10 IT companies in the Philippines and those from the top 10 companies by market capitalization. Feng, Bagheri, Ensan, and Jovanovic (2017)

To mitigate the 128-token limitation of the embedding model (all-mpnet-base-v2) von Platen (2024), fixed-length chunking was employed. Unlike semantic chunking, this method segments text in fixed width length while preserving contextual integrity, ensuring that semantically meaningful units remain intact. By using fixed length chunking, the model maintains a high degree of accuracy in text embeddings, reducing information loss and enhancing representation quality. Zach McCormick (2024)

2.1 NLP Skill Gap Assessment and Visualization Code

Setup

```
import os
import re
import numpy as np
import plotly.express as px
from sklearn.manifold import TSNE
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sentence_transformers import SentenceTransformer, util
import nltk
from plotly.subplots import make_subplots
import pandas as pd
```

Libraries and Modules

- **os**: Provides a way to interact with the operating system, such as reading files from directories.
- **re**: Used for regular expression operations, which are useful for text preprocessing.
- **numpy**: A fundamental package for numerical computations in Python.
- **plotly.express**: A high-level interface for creating interactive plots.
- **sklearn.manifold.TSNE**: Used for dimensionality reduction through t-SNE.
- **nltk.tokenize**: Provides functions for tokenizing text into words and sentences.
- **nltk.corpus.stopwords**: Contains a list of common stopwords that can be filtered out during text preprocessing.
- **nltk.stem.WordNetLemmatizer**: Used for lemmatizing words, reducing them to their base or root form.
- **sentence_transformers**: Provides tools for embedding sentences into high-dimensional vectors.

- **plotly.subplots**: Used for creating complex subplot layouts.
- **pandas**: A powerful data manipulation and analysis library.

Downloading NLTK Resources

```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

For sentence segmentation and word tokenization; Punkt tokenizer was used, this module breaks the text dataset into individual words and sentences, while Stopwords module filters out non informative commonly used English words such as "the", "is", "in", etc. While Wordnet is used to sense tokenized text disambiguation, and lemmatization. These packages simplify the raw dataset sentences.

Text Processing

```
def preprocess_text(text):
    """Preprocesses text by lowercasing, removing non-alphanumeric
    characters, tokenizing, removing stopwords, and lemmatizing."""
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    words = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]
    lemmatizer = WordNetLemmatizer()
    words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(words)
```

The text preprocessing pipeline begins with converting all characters to lowercase to maintain uniformity. Then, non-alphanumeric characters are removed using regular expressions, ensuring that only letters and numbers remain. The cleaned text is then tokenized, breaking it into individual words for semantic analysis. To enhance relevance, stopwords function was used to filter out common words that do not add significant meaning. Finally, lemmatization is applied, reducing words to their base or root form to standardize variations and improve text analysis.

Semantic Chunking

```
def semantic_chunk_text(text, sentences_per_chunk=5):
    """Chunks text into smaller segments of approximately sentences_per_chunk
        sentences."""
    sentences = sent_tokenize(text)
    chunks = [' '.join(sentences[i:i + sentences_per_chunk]) for i in range(0,
        len(sentences), sentences_per_chunk)]
    return chunks
```

This function breaks the text into individual sentences for structured processing. Additionally, the **chunks** variable organizes these sentences into predefined groups based on a set number of sentences per chunk, enabling more efficient text analysis and processing.

Semantic Document Comparison

```
def compare_documents(course_text, listings_text, model):
    """Compares course and listings documents by calculating cosine similarity
        and preparing data for visualization."""
    course_chunks = semantic_chunk_text(course_text)
    listings_chunks = semantic_chunk_text(listings_text)
    course_embeddings = model.encode(course_chunks,
        convert_to_tensor=True).cpu().numpy()
    listings_embeddings = model.encode(listings_chunks,
        convert_to_tensor=True).cpu().numpy()
    cosine_sim = util.cos_sim(course_embeddings.mean(axis=0),
        listings_embeddings.mean(axis=0))
    return cosine_sim.item(), course_embeddings
```

Semantic chunking utilizes the **semantic_chunk_text** function to divide text into meaningful segments while preserving context. These chunks are then embedded into high-dimensional vectors using the SentenceTransformer model. Finally, cosine similarity is computed between the average embeddings of course content and job listings to measure their semantic alignment.

Visualization

```
def create_combined_visualization( all_data , listings_embeddings,
    output_html="combined_semantic_relationships.html"):
    """Creates a combined visualization of semantic relationships for all
        courses and listings ."""
    all_embeddings = np.vstack([embedding for _, embedding in all_data] +
        [embedding for embedding in listings_embeddings])
    n_samples = all_embeddings.shape[0]
    perplexity = min(30, n_samples − 1) # Ensure perplexity is less than the
        number of samples
    tsne = TSNE(n_components=2, perplexity=perplexity, random_state=42)
    reduced_embeddings = tsne.fit_transform(all_embeddings)

    labels = [name for name, _ in all_data] + ['Listings'] *
        len(listings_embeddings)
    colors = ['Course'] * len(all_data) + ['Listings'] *
        len(listings_embeddings)

    df = pd.DataFrame({
        "x": reduced_embeddings[:, 0],
        "y": reduced_embeddings[:, 1],
        "Label": labels,
        "Document": colors
    })

    fig = px.scatter(df, x="x", y="y", color="Document",
        hover_data=["Label"], title="Combined Semantic Relationships")
    fig.write_html(output_html)
    print(f"Combined visualization saved to {output_html}")
```

The process begins with combining embeddings, where all course and job listing embeddings are stacked into a single array. To facilitate visualization, dimensionality reduction is applied using t-SNE, mapping high-dimensional embeddings into a 2D space. The perplexity parameter, which balances local and global structure in t-SNE, is set to at least 30 or n_samples -1 to remain below the total number of samples. Next, a DataFrame is constructed, organizing the reduced embeddings along with their corresponding labels and document types. Finally, Plotting is performed using Plotly, generating an interactive scatter plot to visually explore the relationships between course content and job market expectations.

Execution

```
if __name__ == "__main__":
    listings_file =
        "/workspaces/ITSkillsGap/Skill/top10_PH_IT_Comp_GPC.txt"
    course_dir = "/workspaces/ITSkillsGap/Skill/Docs/Course_Itemized"

    try:
        with open(listings_file, 'r') as file:
            listings_text = file.read()
    except FileNotFoundError:
        print(f"Error: Listings file not found at {listings_file}")
        exit()
    model = SentenceTransformer('jinaai/jina-embeddings-v3')
    listings_chunks = semantic_chunk_text(listings_text)
    listings_embeddings = model.encode(listings_chunks,
        convert_to_tensor=True).cpu().numpy()

    all_data = []
    for filename in os.listdir(course_dir):
        if filename.endswith(".txt"):
            course_file = os.path.join(course_dir, filename)
            course_name = filename[:-4]
            try:
                with open(course_file, 'r') as file:
                    course_text = file.read()
            except FileNotFoundError:
                print(f"Error: Course file not found at {course_file}.
                    Skipping.")
                continue # Skip to the next file
            similarity_score, course_embedding =
                compare_documents(course_text, listings_text, model)

            if similarity_score is not None: # Only add if comparison was
                successful
                all_data.append((course_name, course_embedding))
                print(f"Cosine Similarity between {course_name} and Listings:
                    {similarity_score:.5f}")

    if all_data: # Check if any course data was successfully processed
        create_combined_visualization(all_data, listings_embeddings)
    else:
        print("No course data was successfully processed. Visualization not
            created.")
```

The process begins with reading the listings file, extracting job listing text from the specified source. Next, the SentenceTransformer model (all-mpnet-base-v2) is loaded, selected for its ability to generate high-quality sentence embeddings. The listings text is then processed, chunked, and encoded into embeddings. Similarly, course content is processed by iterating through course files, chunking and encoding the text, and computing cosine similarity between course embeddings and job listings. Finally, if any course data is successfully processed, a visualization is generated and saved, providing insights into the alignment between academic curricula and industry requirements.

2.2 TF-IDF Assessment Code

While semantic analysis offers a high-level visualization of the skill gap between the BSIT curriculum and job listings, it does not explicitly highlight specific discrepancies. To bridge this gap, Term Frequency-Inverse Document Frequency (TF-IDF) was employed to detect lexical mismatches between industry expectations and academic instruction. The ranked terms, alongside their similarity scores, were then qualitatively examined to extract key insights.

Setup

```
import os
import re
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import nltk

# Download required NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

Libraries and Modules

- **os**: Provides a way to interact with the operating system, such as reading files from directories.
- **re**: Used for regular expression operations, which are useful for text preprocessing.
- **numpy**: A fundamental package for numerical computations in Python.
- **pandas**: Data manipulation and analysis library.
- **sklearn.feature_extraction.text.TfidfVectorizer**: Used for calculating TF-IDF scores for terms in the text.
- **nltk.tokenize**: Provides functions for tokenizing text into words.
- **nltk.corpus.stopwords**: Contains a list of common stopwords that can be filtered out during text preprocessing.

- **nltk.stem.WordNetLemmatizer**: Used for lemmatizing words, reducing them to their base or root form.

Text Proecessing

```
def preprocess_text(text):
    """Preprocesses text by lowercasing, removing non-alphanumeric
        characters, tokenizing, removing stopwords, and lemmatizing."""
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    words = word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]
    lemmatizer = WordNetLemmatizer()
    words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(words)
```

The text preprocessing pipeline begins with lowercasing, converting all characters to lowercase for consistency. Next, non-alphanumeric characters are removed using regular expressions, ensuring only letters and numbers remain. The text is then tokenized, splitting it into individual words. Common stopwords that do not add significant meaning are filtered out, reducing noise. Finally, lemmatization is applied to transform words into their base or root form, enhancing uniformity across variations of the same word.

Skill Extraction

```
def extract_skills(text):
    """Extracts skills and qualifications from text using TF-IDF."""
    vectorizer = TfidfVectorizer(stop_words='english')
    tfidf_matrix = vectorizer.fit_transform([text])
    feature_names = vectorizer.get_feature_names_out()
    scores = tfidf_matrix.toarray().flatten()
    skill_scores = dict(zip(feature_names, scores))
    return skill_scores
```

TF-IDF Vectorization is performed using TfidfVectorizer, which computes the Term Frequency-Inverse Document Frequency (TF-IDF) scores for words in the corpus. From this, skill extraction is carried out by identifying terms with their respective TF-IDF scores, returning a structured dictionary that highlights the most relevant skills based on their significance in the dataset.

Skill Gap Identification

```
def identify_skill_gaps(job_postings, curriculum):
    """Identifies skill gaps between job postings and curriculum."""
    job_postings_text = ' '.join(job_postings)
    curriculum_text = ' '.join(curriculum)
    job_postings_skills = extract_skills(preprocess_text(job_postings_text))
    curriculum_skills = extract_skills(preprocess_text(curriculum_text))

    skill_gaps = {skill: job_postings_skills[skill] for skill in
        job_postings_skills if skill not in curriculum_skills}

    return skill_gaps
```

The text from job postings and curriculum is first combined into single strings to facilitate analysis. Then, the extract_skills function is applied to identify relevant skills from both datasets based on their TF-IDF scores. Finally, the process identifies skill gaps by comparing the extracted skills, highlighting those present in job postings but absent in the curriculum, and returning them as a dictionary with corresponding significance scores.

Execution

```
if __name__ == "__main__":
    listings_file =
        "/workspaces/ITSkillsGap/Skill/top10_PH_IT_Comp_GPC.txt"
    course_dir = "/workspaces/ITSkillsGap/Skill/Docs/Course_Itemized"
    try:
        with open( listings_file , 'r') as file:
            listings_text = file.read()
    except FileNotFoundError:
        print(f"Error: Listings file not found at { listings_file }")
        exit ()
    job_postings = [ listings_text ]
    curriculum = []
    for filename in os. listdir (course_dir):
        if filename.endswith(".txt"):
            course_file = os.path.join(course_dir, filename)
            try:
                with open(course_file, 'r') as file:
                    curriculum.append(file.read())
            except FileNotFoundError:
                print(f"Error: Course file not found at { course_file }.
                    Skipping.")
                continue # Skip to the next file
    skill_gaps = identify_skill_gaps (job_postings, curriculum)
    # Save the results to a CSV file
    output_file = "/workspaces/ITSkillsGap/result.csv"
    skill_gaps_df = pd.DataFrame(skill_gaps.items(), columns=['Skill', 'Score'])
    skill_gaps_df .to_csv( output_file , index=False)
    print(f" Identified skill gaps have been saved to { output_file }")
```

The job postings text is first read from the specified file, followed by an iteration through course files in the designated directory to extract curriculum content. The identify_skill_gaps function then analyzes discrepancies between the required job skills and the curriculum offerings. Finally, the identified skill gaps are saved to a CSV file for further examination and analysis.

This data processing method is employed to qualitatively evaluate the skill gap between the academic curriculum and industry requirements. Through natural language processing, job listings and curriculum datasets are processed at scale, enabling structured assessment. Additionally, Gemini Pro LLM is utilized to standardize source data, ensuring consistency and reducing variations in textual representations

3 Results

This section presents the results of the semantic analysis comparing the BSIT curriculum in the Philippines with job listings from the country's top IT companies and the largest corporations by market capitalization. The analysis evaluates the alignment between academic coursework and industry requirements, identifying key skill gaps that may affect graduate employability.

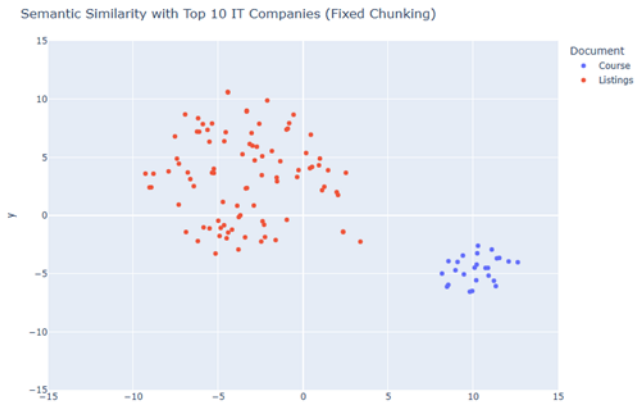

Fig. 1: 2D Clustering of courses and IT related job listings by semantic similarity

Fig. 2: 3D clustering of courses and IT related job listings by semantic similarity

The visualization highlights a greater semantic distance between IT job listings and the BSIT curriculum compared to the clustering of academic courses. The X-Y axis represents the semantic distance between datasets. Notably, the curriculum data points are more tightly grouped, reflecting the standardized course outcomes mandated by CHED. This cohesion is further reinforced by the ladderized structure of the curriculum, where courses within the same domain are designed as prerequisites for one another, leading to greater semantic similarity.

In contrast, job listings are treated as a single aggregated document, where individual postings are combined and segmented into fixed-length chunks (128 tokens). Semantic comparison is performed after chunking, with each N-sized segment evaluated against the curriculum data to assess alignment.

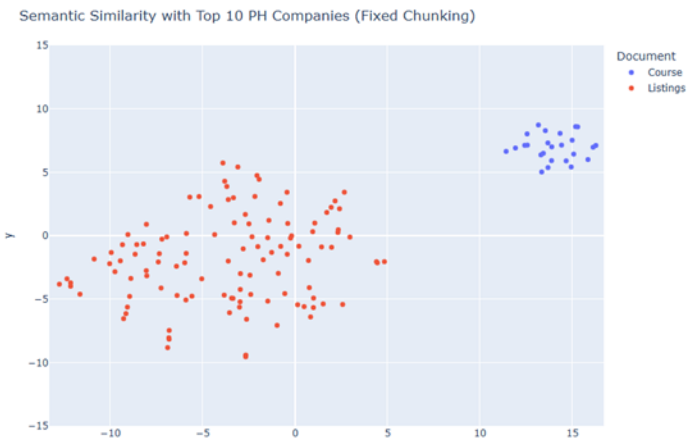

Fig. 3: 2D clustering of courses and top 10 PH companies job listings by semantic similarity.

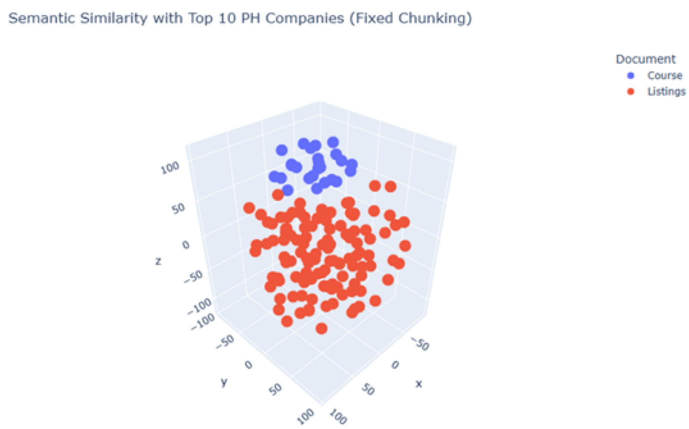

Fig. 4: 3D clustering of courses and top 10 PH companies job listings by semantic similarity.

Skills acquired in the BSIT curriculum align more closely with the requirements of IT-specific job listings but show greater divergence from those of the top 10 companies by market capitalization. However, in both cases, the curriculum does not fully bridge the identified skill gap. The narrower spread of IT job listings suggests a more consistent yet broad set of industry demands, whereas the wider distribution of top 10 company listings indicates a more varied skill expectation.

The top 10 companies by market capitalization exhibit a broader distribution compared to IT-specific job listings. Notably, the dataset for these companies includes only IT-related positions, ensuring alignment with the expected competencies of BSIT graduates. The horizontal spread in the visualization suggests a wider range of skill requirements. Meanwhile, the IT curriculum is seven points further on the 2D X-axis vector graph.

Top 10 PH IT Companies (from GPW)		**Top 10 PH Companies by Market Cap**	
Course	**Cosine Similarity**	**Course**	**Cosine Similarity**
NET101	0.51672	NET101	0.59757
NET102	0.50107	NET102	0.53467
HCI101	0.45472	HCI101	0.50185
SA101	0.51648	SA101	0.68958
HCI102	0.44669	HCI102	0.48637
IPT102	0.50657	IPT102	0.56804
CAP102	0.53942	CAP102	0.62610
CC106	0.56224	CC106	0.61517
MS101	0.48899	MS101	0.58527
SIA102	0.52434	SIA102	0.60690
CC105	0.55925	CC105	0.60404
WS101	0.47666	WS101	0.60285
CC101	0.40365	CC101	0.49490
PT101	0.49560	PT101	0.63930
CAP101	0.51488	CAP101	0.62214
PF101	0.46572	PF101	0.51720
CC103	0.44253	CC103	0.51801
IM101	0.49467	IM101	0.58704
MS102	0.49921	MS102	0.55072
CC104	0.47773	CC104	0.52031
SIA101	0.50057	SIA101	0.60272
IPT101	0.51291	IPT101	0.67862
IAS101	0.50719	IAS101	0.61689
SP101	0.51846	SP101	0.52521
CC102	0.49670	CC102	0.58137
Average	**0.4969188**	**Average**	**0.5789136**

Table 4: Cosine Similarity of BSIT Courses to Industry Requirements

The cosine similarity analysis indicates that the BSIT curriculum exhibits a higher average similarity with job listings from the Top 10 Philippine companies (0.57) than with IT-specific job postings (0.49). However, the t-SNE visualizations suggest a closer structural relationship between the curriculum and IT-focused listings. This apparent discrepancy arises due to the fundamental differences in how cosine similarity and t-SNE interpret textual embeddings. Cosine similarity quantifies direct lexical overlap, favoring non-IT companies whose job descriptions incorporate more generalized IT-related terminology. Conversely, IT-focused job listings tend to feature specialized, domain-specific language, which reduces their textual similarity with the curriculum despite their closer alignment in structural representation. In contrast, t-SNE prioritizes local clustering over absolute distances, resulting in a visualization where curriculum embeddings remain in closer proximity to IT listings.

Further, variance in job descriptions contributes to this observation. Non-IT firms require a broader range of competencies, leading to higher per-course similarity scores. IT-specific roles demand more specialized knowledge, lowering their direct textual similarity yet maintaining structural closeness. Despite these findings, both similarity scores remain below 0.8 (0.49 and 0.57), indicating that the curriculum does not fully align with industry demands. Addressing this gap necessitates curricular refinements that integrate both per-course textual relevance and overall structural coherence to enhance alignment with evolving industry requirements.

PH IT TF-IDF			**PH Comp TF-IDF**		
Rank	Keyword	Score	Rank	Keyword	Score
1	aboveaverage	0.01478	1	customer	0.05106
1	accelerate	0.01478	2	excellent	0.04425
2	acceptable	0.01430	3	analytics	0.04198
2	accessible	0.01430	4	ai	0.03933
2	accomplishing	0.01430	5	delivery	0.03063
2	accountability	0.01430	6	industry	0.02761
3	accounting	0.01382	6	initiative	0.02761
3	accountrelated	0.01382	7	governance	0.02647
4	accredited	0.01335	8	sap	0.02496
4	accuracy	0.01335	9	internal	0.02420
5	achieve	0.01239	10	insight	0.02345
6	acquisition	0.01192	10	unit	0.02345
6	actionable	0.01192	11	crossfunctional	0.02231
7	adapt	0.01144	11	lead	0.02231
8	adaptability	0.01096	11	provide	0.02231
8	adaptable	0.01096	12	pipeline	0.02080
9	adherence	0.01049	13	sale	0.02042
9	adhoc	0.01049	14	leadership	0.02004
9	adobe	0.01049	15	marketing	0.01853
9	adopt	0.01049	15	operational	0.01853
10	advantage	0.01001	16	aws	0.01777
10	advise	0.01001	16	expertise	0.01777
10	advocating	0.01001	16	supporting	0.01777
11	aggregate	0.00953	17	visualization	0.01740

Continued on next page

Continued from previous page

PH IT TF-IDF			PH Comp TF-IDF		
Rank	Keyword	Score	Rank	Keyword	Score
11	agilescrum	0.00953	18	financial	0.01664
11	ai	0.00953	19	cicd	0.01437
12	aimlpowered	0.00906	19	dashboard	0.01437
13	angularjs	0.00810	20	cybersecurity	0.01324
13	animate	0.00810	21	azure	0.01248

Table 5: Top 30 Keywords by TF-IDF Score

The TF-IDF analysis highlights key industry-relevant terms in both the general Philippine corporate sector and the IT industry. In the PH Comp TF-IDF results, terms such as *customer, excellent, analytics*, and *AI* suggest a strong emphasis on customer satisfaction, data-driven decision-making, and emerging technologies. (*Table 5*) Meanwhile, the PH IT TF-IDF results contain terms such as *AI, agilescrum, adopt*, and *advocating*, which indicate a focus on adaptability, modern software development practices, and AI-driven solutions. The overlap in keywords like AI and analytics underscores the increasing role of technology across industries, reinforcing the need for IT professionals to align with business objectives and industry expectations.

4 Discussion

Based on the collected data and the subsequent semantic and qualitative analysis, this study finds that the BSIT curriculum in the Philippines, structured around CMO No. 25, Series of 2015 by the Commission on Higher Education (CHED), remains insufficient in bridging the identified skill gap. The findings indicate that despite nearly a decade of implementation, the curriculum has yet to fully equip BSIT graduates with the competencies required to remain competitive in both IT-specific and non-IT industries within the Philippines. This persistent misalignment has impacted thousands of BSIT graduates since 2015-2024 (292,373) Statista (2024c), leaving them with deficiencies in key competencies necessary for the evolving job market. The curriculum's coverage of industry-relevant skill sets remains inadequate, resulting in graduates who are not only less competitive but also under-prepared in critical subject areas. Addressing this gap requires a strategic revision of the curriculum, ensuring that BSIT students acquire both foundational and specialized skills aligned with current industry demands.

Acknowledgements. The author expresses gratitude to the institutions and professionals who provided insights and references that guided this study. This work is dedicated to the tireless students of Computing studies in the Philippines.

Declarations

Funding
This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Data availability
The supporting dataset used in this study is openly available at https://osf.io/rdh8p, and the data analysis plan can be accessed at https://osf.io/k4jxf.

Materials availability
Not applicable.

Code availability
Not applicable.

References

9th Congress of the Philippines (1994). *Higher education act of 1994.* https://lawphil.net/statutes/repacts/ra1994/ra_7722_1994.html.

Alcala, A.C. (2015). *Updated policies and standards for information technology education (it)* (No. 25). Pasig City, Philippines: Commision on Higher Education.

Angelo, A. (2021). *Skills mismatch hampers philippine economic growth, experts say* (Tech. Rep.). Philippine Institute for Development Studies.

Bringula, R. (2016). Employable skills of information technology graduates in the philippines: Do industry practitioners and educators have the same view? *The 21st Western Canadian Conference*, 1–6, https://doi.org/10.1145/2910925.2910928

Buelva, A. (2002, October 18). *Big bang for big blue.* https://www.philstar.com/business/technology/2002/10/18/180361/big-bang-big-blue. (Accessed: 2025-02-12)

Commission on Higher Education (1998). *Policies and standards for associate in computer technology program (act)* (No. 5). Pasig City, Philippines: Commission on Higher Education. Retrieved from https://nlpdl.nlp.gov.ph/CH01/MO/1998/CH0MO0005s1998.pdf (Accessed: 2025-02-12)

Commission on Higher Education (2006). *Policies and standards for information technology education (ite) programs* (No. 53). Manila City, Philippines: Commission on Higher Education. Retrieved from https://fs2.mapua.edu.ph/2020QS/Other%20Evidences/CQIO/BSCS%20Certificates/CHED-CMOs/CMO-No.53-s2006.pdf (Accessed: 2025-02-12)

Commission on Higher Education (2015). *Ched memorandum order no. 25, series of 2015: Revised policies, standards, and guidelines for bachelor of science in computer science (bscs), bachelor of science in information systems (bsis), and bachelor of science in information technology (bsit)* (No. 25). Quezon City, Philippines: Commission on Higher Education. Retrieved from https://www.pacu.org.ph/wordpress/wp-content/uploads/2017/06/CMO-No.25-s2015.pdf (Accessed: 2025-02-12)

CompaniesMarketCap (2024, 11). *Largest companies of the philippines by market capitalization.* https://web.archive.org/web/20250214023640/https://companiesmarketcap.com/philippines/largest-companies-in-the-philippines-by-market-cap/. (Accessed: 2024-11-15)

Data Commons (2024). *Proportion of individuals using the internet.* https://datacommons.org/browser/sdg/IT_USE_ii99. (Accessed: 2025-02-14)

Feng, Y., Bagheri, E., Ensan, F., Jovanovic, J. (2017). The state of the art in semantic relatedness: a framework for comparison. *The Knowledge Engineering Review*, *32*, e10, https://doi.org/10.1017/S0269888917000029

Great Place to Work Philippines (2024). *Philippines best workplaces in it-bpm 2024 archives. great place to work philippines* (Tech. Rep.). Great Place to Work. https://greatplacetowork.com.ph/awards-list/philippines-best-workplaces-in-it-bpm-2024/#phbwitbpm2024.

International Business Machines (IBM) (n.d.). *Ibm archives: Ibm 650 introduction 2.* https://web.archive.org/web/20230306131515/https://www.ibm.com/ibm/history/exhibits/650/650_intro2.html. (Accessed: 2025-02-12)

IT & Business Process Association of the Philippines (IBPAP) (2019). *The philippine it-bpm industry growth forecast.* https://www.ibpap.org/knowledge-hub/research. (Accessed: 2025-02-14)

Lallana EC, S.C. (2007). *Towards universal internet access. ideacorp and intel technology philippines.* ideCorp.

Quisumbing, L.R. (1998). *Policies and standards for degree programs in computer science* (No. 7). Manila City, Philippines: Department of Education Culture and Sports.

Statista (2024a, December). *Internet penetration rate in the philippines from 2020 to 2029.* https://www.statista.com/statistics/975072/internet-penetration-rate-in-the-philippines/. (Accessed: 2025-02-14)

Statista (2024b, December). *Mobile phone users philippines 2021-2029.* https://www.statista.com/forecasts/558756/number-of-mobile-internet-user-in-the-philippines. (Accessed: 2025-02-14)

Statista (2024c, August). *Number of higher education graduates of it-related disciplines in the philippines from academic year 2014/15 to 2021/22.* https://www.statista.com/statistics/1312885/philippines-number-of-higher-education-graduates-of-it-related-disciplines/. (Accessed: 2025-02-20)

von Platen, P. (2024, 1). *All mpnet base v2.* https://huggingface.co/sentence-transformers/all-mpnet-base-v2/. (Accessed: 2025-02-15)

Zach McCormick (2024). *Solving the out-of-context chunk problem for rag.* https://d-star.ai/solving-the-out-of-context-chunk-problem-for-rag. (Accessed: 2025-02-15)