TTDS Coursework 2 Report

IR Evaluation, Text Analysis, and Classification

Student ID: s2414220

December 2, 2025

1 Implementation Overview

This coursework involves three main components: IR system evaluation, text corpus analysis, and sentiment classification. All implementations were completed in Python using standard libraries including pandas, numpy, scikit-learn, and NLTK.

1.1 Code Structure

The implementation is organized into three main classes:

- `IREvaluator`: Computes precision, recall, R-precision, AP, and nDCG metrics for IR systems
- `TextAnalyzer`: Performs mutual information, chi-square analysis, and LDA topic modeling
- `SentimentClassifier`: Implements baseline and improved sentiment classification models

1.2 Key Implementation Challenges

Several challenges were encountered during implementation:

1. **nDCG Calculation**: Ensuring the correct formula from Lecture 9 was used, particularly handling the log base and position indexing correctly (i.e., $\text{rel}_i / \log_2(i)$ for $i \geq 2$).
2. **Data Structure Design**: Converting qrels into an efficient dictionary format (`query_id` $\rightarrow$ `doc_id` $\rightarrow$ `relevance`) for O(1) lookup time.
3. **Binary vs. Graded Relevance**: Correctly distinguishing between metrics requiring binary relevance (P, R, AP) versus graded relevance (nDCG).

1.3 What Was Learned

This coursework provided hands-on experience with:

- Implementation of standard IR evaluation metrics from first principles
- Statistical significance testing for comparing system performance
- Feature selection techniques (MI and χ^2) for corpus comparison
- Topic modeling with LDA and interpretation of results
- Practical challenges in sentiment classification and model improvement

2 IR Evaluation Results

2.1 System Performance Comparison

Table 1 presents the mean performance of all six IR systems across the ten test queries, evaluated using six different metrics.

Table 1: Mean Performance of IR Systems Across All Metrics

System	P@10	R@50	R-Prec	AP	nDCG@10	nDCG@20
1	0.390	0.834	0.401	0.400	0.363	0.485
2	0.220	**0.867**	0.252	0.300	0.200	0.246
3	**0.410**	0.767	**0.449**	**0.451**	**0.420**	**0.511**
4	0.080	0.189	0.049	0.075	0.069	0.076
5	**0.410**	0.767	0.358	0.364	0.332	0.424
6	**0.410**	0.767	**0.449**	0.445	0.400	0.490

2.2 Best Performing Systems by Metric

Table 2 summarizes the best and second-best systems for each metric, along with statistical significance test results using a two-tailed t-test ($\alpha = 0.05$).

Table 2: Statistical Significance Analysis of Best Performing Systems

Metric	Best Sys	Mean	2nd Sys	Mean	p-value
P@10	3 (tie with 5,6)	0.410	5	0.410	1.000
R@50	2	0.867	1	0.834	0.703
R-Precision	3 (tie with 6)	0.449	6	0.449	1.000
AP	3	0.451	6	0.445	0.967
nDCG@10	3	0.420	6	0.400	0.883
nDCG@20	3	0.511	6	0.490	0.868

2.3 Analysis and Discussion

Overall Best System: System 3 emerges as the strongest performer, achieving the highest scores in five out of six metrics (P@10, R-Precision, AP, nDCG@10, nDCG@20). System 2 performs best only on R@50, suggesting it retrieves more relevant documents in the top 50 results.

Statistical Significance: Notably, *none of the best systems are statistically significantly better than their second-place counterparts* (all p-values > 0.05). This can be attributed to several factors:

1. **Small Sample Size**: With only 10 queries, the statistical power is limited. Small variations in performance across queries lead to large standard deviations, making it difficult to detect significant differences.

2. **Tied Performances**: For P@10 and R-Precision, Systems 3, 5, and 6 achieved identical or near-identical mean scores (e.g., all three systems scored 0.410 on P@10), resulting in p-values of 1.000.

3. **High Variance**: IR system performance often varies considerably across different queries depending on query difficulty and relevance distribution. This natural variance obscures small performance differences between systems.

4. **Similar Retrieval Quality**: Systems 3 and 6 show remarkably similar performance patterns across all metrics, suggesting they may employ similar retrieval strategies or ranking functions.

Practical Implications: While statistical significance is not achieved, System 3 consistently achieves the highest or near-highest scores across multiple metrics, suggesting it is the most robust choice in practice. The lack of significance primarily reflects limitations in test collection size rather than absence of real performance differences.

Metric-Specific Observations:

- System 2's superiority in R@50 but poor performance in precision-oriented metrics (P@10, nDCG) suggests a recall-focused strategy that retrieves many relevant documents but with lower ranking quality.
- System 4 performs poorly across all metrics, indicating fundamental issues with its retrieval approach.
- The strong correlation between systems' AP, nDCG@10, and nDCG@20 scores suggests these metrics capture similar aspects of ranking quality.

3 Text Analysis

3.1 Corpus Overview

The analysis focused on three religious text corpora: the Quran, Old Testament (OT), and New Testament (NT). Each verse was treated as a separate document. Preprocessing included tokenization, lowercasing, and stopword removal.

3.2 Mutual Information and Chi-Square Analysis

3.2.1 Top Features by Mutual Information

Table 3: Top 10 Tokens by Mutual Information Score

Quran		**Old Testament**		**New Testament**	
Token	**MI**	**Token**	**MI**	**Token**	**MI**
bargain	2.576	overflows	0.690	eunice	2.236
trunks	2.576	circumference	0.690	infallible	2.236
needlessly	2.576	ishpan	0.690	bethphage	2.236
unsuccessful	2.576	embalm	0.690	rigid	2.236
vicious	2.576	dismayed	0.690	murmuring	2.236
kinsmen	2.576	shedder	0.690	apelles	2.236
evert	2.576	musician	0.690	conversion	2.236
mim	2.576	defer	0.690	pilot	2.236
insignificant	2.576	gluttons	0.690	parthians	2.236
aimlessly	2.576	treading	0.690	abba	2.236

3.2.2 Top Features by Chi-Square

3.2.3 Comparison of MI and χ^2 Rankings

The two feature selection methods reveal strikingly different characteristics of the corpora:

Mutual Information (MI) identifies *corpus-exclusive vocabulary* but exhibits a known limitation: all top-ranked terms within each corpus share identical scores (2.576 for Quran, 0.690

Table 4: Top 10 Tokens by χ^2 Score

Quran		Old Testament		New Testament	
Token	χ^2	**Token**	χ^2	**Token**	χ^2
muhammad	1852.1	shall	1504.9	jesus	3026.7
god	1792.5	lord	1114.1	christ	1764.5
certainly	1682.7	israel	1096.0	disciples	741.0
believers	1588.1	king	862.0	things	673.9
torment	1381.9	land	471.6	paul	529.0
unbelievers	874.5	sons	423.4	peter	529.0
revelations	814.4	judah	402.0	john	408.2
guidance	810.9	house	377.8	spirit	374.9
messenger	793.1	david	323.7	gospel	300.3
quran	753.0	hand	280.2	grace	298.4

for OT, 2.236 for NT). This occurs because these words appear exclusively in their respective corpus, achieving the theoretical maximum information gain of $\log_2(N/N_c)$ where N is total documents and N_c is corpus size. However, these terms are predominantly rare proper nouns (e.g., "ishpan", "apelles") or low-frequency words with limited semantic salience, making MI rankings less interpretable for understanding corpus themes.

Chi-Square (χ^2) produces more semantically meaningful rankings by prioritizing *high-frequency discriminative terms*. The top words clearly capture each corpus's thematic identity:

- **Quran**: "muhammad", "believers", "torment", "messenger" — reflecting Islamic theology and prophetic discourse
- **Old Testament**: "israel", "king", "judah", "david" — emphasizing Hebrew monarchy and nationhood
- **New Testament**: "jesus", "christ", "disciples", "grace" — centering on Christian soteriology

Key Difference: MI measures *lexical uniqueness* (perfect discrimination even for singletons), while χ^2 tests *statistically significant deviation* from expected distribution, inherently favoring terms with sufficient occurrence counts. For thematic corpus analysis, χ^2 proves more robust, as its rankings reflect content-defining keywords rather than incidental vocabulary.

3.3 LDA Topic Modeling

An LDA model with 20 topics was trained on all verses from the three corpora. For each corpus, the most prominent topic (highest average document-topic probability) was identified.

Table 5: Most Prominent Topics and Top Tokens for Each Corpus

Corpus	**Topic ID (Avg Score)**	**Top 10 Tokens**
Quran	Topic 19 (0.358)	god, people, would, say, one, muhammad, certainly, lord, torment, know
Old Testament	Topic 16 (0.080)	king, came, david, saying, said, sent, lord, people, jerusalem, house
New Testament	Topic 10 (0.143)	things, jesus, said, life, christ, answered, god, world, one, come

3.3.1 Topic Labels

Based on the top tokens, I assign the following interpretive labels to these topics:

- **Quran Topic 19**: "Divine Guidance and Prophethood" — dominated by theological terminology (god, lord, muhammad, torment) reflecting Islamic monotheism and prophetic teachings
- **Old Testament Topic 16**: "Davidic Monarchy and Temple" — centered on Hebrew kingship (king, david, jerusalem, house) representing the historical-political narrative of ancient Israel
- **New Testament Topic 10**: "Christ's Life and Teachings" — focused on Jesus's ministry (jesus, christ, life, things, world) capturing the Gospel narrative and theological discourse

3.3.2 LDA Insights and Comparison with MI/χ^2

Thematic Coherence Varies Dramatically: The average topic scores reveal striking differences in corpus homogeneity:

- **Quran (0.358)**: Exhibits highest thematic coherence, with over one-third of its content concentrated in a single topic. This reflects the Quran's unified theological focus on monotheism and prophethood.
- **New Testament (0.143)**: Moderate coherence, balancing Gospel narratives with diverse epistles and apocalyptic literature.
- **Old Testament (0.080)**: Lowest coherence, consistent with its encyclopedic nature spanning history (Genesis-Kings), law (Leviticus), wisdom (Proverbs), and prophecy (Isaiah).

Cross-Corpus Topic Patterns: Analysis of the full 20-topic distribution reveals:

- **Corpus-Exclusive Topics**: Topic 19 is Quran-specific (0.358 vs. 0.018 in OT), while Topic 10 is NT-dominant (0.143 vs. 0.024 in OT). These represent each text's unique theological identity.
- **Biblical Continuity**: Topics 5-7 show OT dominance (0.07-0.08) with moderate NT presence (0.03-0.05) but low Quran scores (0.01-0.04). These likely capture prophetic/legal themes shared between Old and New Testaments, reflecting theological continuity absent in the Quran.
- **OT Dispersion**: The Old Testament scores moderately (0.04-0.08) across Topics 3-9, confirming its diverse compositional structure.

LDA vs. MI/χ^2: The methods provide complementary insights:

- **MI/χ^2** identify *discriminative vocabulary* — individual words that distinguish corpora (e.g., "muhammad" vs. "jesus").
- **LDA** reveals *latent thematic structure* — co-occurring word patterns that form semantic topics (e.g., "king + david + jerusalem + house" forming a monarchy theme).
- **Key Advantage of LDA**: Captures *polysemy* and *context*. For example, "lord" appears in all corpora but within different semantic contexts (Quran: divine authority; OT: covenant relationship; NT: christological title). LDA models these contextual differences through topic distributions, while MI/χ^2 treat each word atomically.

4 Text Classification

4.1 Dataset and Experimental Setup

The sentiment classification task involved three-way classification (positive, negative, neutral) of tweets. The training data was split 90/10 into training and development sets. All experiments used the same random seed for reproducibility.

4.2 Baseline System

The baseline system followed the lab 7 approach:

- **Preprocessing**: Basic tokenization, lowercasing
- **Features**: Bag-of-words (BOW) with `CountVectorizer`
- **Classifier**: Linear SVM with $C = 1000$

Table 6: Baseline System Performance

Split	P-Pos	R-Pos	F-Pos	P-Neg	R-Neg	F-Neg	P-Neu	R-Neu	F-Neu	P-Macro	F
Train	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	
Dev	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	
Test	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	

4.3 Error Analysis

Three misclassified examples from the development set were analyzed:

1. **Example 1**: [Text]
   Predicted: X, Actual: Y
   Hypothesis: [Why it was misclassified]
2. **Example 2**: [Text]
   Predicted: X, Actual: Y
   Hypothesis: [Why it was misclassified]
3. **Example 3**: [Text]
   Predicted: X, Actual: Y
   Hypothesis: [Why it was misclassified]

4.4 Improved System

Based on error analysis and experiments, the following improvements were implemented:

4.4.1 Improvements Made

1. **[Improvement 1]**: [Description, e.g., "N-gram features (unigrams + bigrams)"]
   - *Motivation*: [Why you tried this]
   - *Implementation*: [How you implemented it]
   - *Result*: [Impact on performance]
2. **[Improvement 2]**: [Description]

- *Motivation*: [Why you tried this]
- *Implementation*: [How you implemented it]
- *Result*: [Impact on performance]

3. **[What Didn't Work]**: [Description of failed attempts]

Table 7: Improved System Performance

Split	P-Pos	R-Pos	F-Pos	P-Neg	R-Neg	F-Neg	P-Neu	R-Neu	F-Neu	P-Macro	F
Train	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	
Dev	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	
Test	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	0.xxx	

4.5 Performance Gains

- **Development Set**: Macro-F1 improved from 0.xxx to 0.xxx (+X.xxx gain)
- **Test Set**: Macro-F1 improved from 0.xxx to 0.xxx (+X.xxx gain)

4.6 Analysis of Dev vs. Test Performance

[Your analysis here - if test performance is lower than dev, discuss possible overfitting; if similar, discuss generalization]

5 Conclusion

This coursework provided comprehensive experience in three core areas of text and data mining. Key takeaways include:

- The importance of statistical testing when comparing system performance
- Different perspectives provided by discriminative (MI/χ^2) vs. generative (LDA) approaches to text analysis
- The iterative nature of improving classification systems through error analysis and experimentation