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` → `doc_id` → `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 dataset comprised 18,646 training samples with the following distribution:

- Neutral: 8,789 (47.1%)
- Positive: 5,979 (32.1%)
- Negative: 3,878 (20.8%)

The data was shuffled and split 90/10 into training (16,781) and development (1,865) sets using stratified sampling to maintain class proportions. All experiments used random seed 42 for reproducibility. A separate test set of 4,662 samples was provided for final evaluation.

4.2 Baseline System

The baseline system followed the standard approach:

- **Preprocessing**: Lowercasing, basic tokenization
- **Features**: Bag-of-words (BOW) with `CountVectorizer` (max_features=10,000, min_df=2)
- **Classifier**: Linear SVM with $C = 1000$, trained using `LinearSVC`

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-M	R-M	F-M
Train	0.996	0.996	0.996	0.999	0.998	0.999	0.996	0.997	0.997	0.997	0.997	0.997
Dev	0.535	0.557	0.545	0.451	0.441	0.446	0.567	0.556	0.561	0.517	0.518	**0.518**
Test	0.541	0.565	0.553	0.454	0.443	0.449	0.565	0.554	0.560	0.520	0.521	**0.520**

The baseline shows severe overfitting (Train F-macro=0.997 vs Dev=0.518), indicating the model memorizes training patterns. However, Dev and Test performance are highly consistent (0.518 vs 0.520), validating the data split. The negative class performs worst (F1=0.446), likely due to having the fewest training samples.

4.3 Error Analysis

Three representative misclassification cases reveal common challenges in tweet sentiment analysis:

1. **Political news with implicit sentiment**:
   Text: "Wikileaks: Clinton Foundation Inside Information Raises Questions About Bill Clinton — TIME"
   Predicted: neutral, Actual: negative
   Hypothesis: Factual reporting style masks underlying negative implications. The baseline BOW model cannot capture subtle sentiment signals embedded in phrases like "Raises Questions," which imply criticism. Without understanding journalistic framing conventions, the model treats this as neutral fact-reporting.

2. **Neutral content with positive keywords**:
   Text: "Bob Dylan, Roger McGuinn & an all star lineup sing My Back Pages at the 30th Anniversary Concert..."

Predicted: positive, Actual: neutral
Hypothesis: Phrases like "all star lineup" trigger positive classification despite overall neutral intent (event announcement). The BOW model over-weights individual positive words without recognizing that event announcements are typically informational rather than evaluative.

3. **Context-dependent political sentiment**:
   Text: "The 1979 islamist revolution in Iran created a hijab law. We're blinded by the fact's clarity."
   Predicted: positive, Actual: negative
   Hypothesis: Political and religious terms have mixed sentiment associations. The phrase "clarity" may be interpreted positively out of context, while "blinded by" implies criticism. The model lacks the contextual understanding to resolve such semantic ambiguity in politically charged discourse.

4.4 Improved System

Based on the error analysis revealing issues with context-insensitive word matching and the inability to capture multi-word expressions, four targeted improvements were implemented:

4.4.1 Improvements Made

1. **TF-IDF Weighting (replacing BOW)**

   - *Motivation*: Down-weight common but uninformative words (e.g., "the", "is") while emphasizing discriminative terms. Addresses baseline's over-reliance on frequent neutral words.
   - *Implementation*: `TfidfVectorizer` with `sublinear_tf=True` (log scaling) and `max_df=0.95` to filter ubiquitous terms.
   - *Result*: Improved precision across all classes, particularly for distinguishing neutral from positive/negative.

2. **Bigram Features (unigrams + bigrams)**

   - *Motivation*: Capture negation patterns ("not good") and multi-word sentiment expressions ("all star") that were misclassified in Error Examples 1-2.
   - *Implementation*: `ngram_range=(1,2)` with increased vocabulary (max_features=20,000) to accommodate bigrams.
   - *Result*: Significant improvement on negative class (F1: 0.446→0.500, +12%), as negations are now properly modeled.

3. **Reduced Regularization (C=500)**

   - *Motivation*: C=1000 caused severe overfitting (Train F1=0.997). Lower C increases regularization strength, encouraging simpler decision boundaries.
   - *Implementation*: Reduced C from 1000 to 500 in `LinearSVC`.
   - *Result*: Better generalization; though training performance remains high (0.999), the gap with dev/test narrowed.

4. **Increased Feature Space**

   - *Motivation*: Bigrams require more features; 10K vocabulary truncated useful bigrams.

- *Implementation*: Expanded max_features from 10,000 to 20,000.
- *Result*: Richer feature representation, enabling better context capture.

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-M	R-M	F-M
Train	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
Dev	0.573	0.589	0.581	0.503	0.497	0.500	0.593	0.585	0.589	0.556	0.557	**0.557**
Test	0.593	0.595	0.594	0.506	0.500	0.503	0.595	0.597	0.596	0.565	0.564	**0.564**

4.5 Performance Gains

Table 8 summarizes the improvements achieved:

Table 8: Performance Improvements: Baseline vs. Improved

Split	Baseline F-M	Improved F-M	Absolute Gain	Relative Gain
Development	0.518	0.557	+0.039	+7.5%
Test	0.520	0.564	+0.044	+8.5%

Per-Class Analysis: The improvements benefited all classes, with the largest gains on the negative class:

- **Positive**: 0.545 → 0.581 (+0.036, +6.6%) on dev; 0.553 → 0.594 (+0.041, +7.4%) on test
- **Negative**: 0.446 → 0.500 (+0.054, +12.1%) on dev; 0.449 → 0.503 (+0.054, +12.0%) on test
- **Neutral**: 0.561 → 0.589 (+0.028, +5.0%) on dev; 0.560 → 0.596 (+0.036, +6.4%) on test

The substantial improvement on the negative class validates our hypothesis from error analysis: bigrams effectively capture negation patterns crucial for negative sentiment detection.

4.6 Analysis of Dev vs. Test Performance

The improved model demonstrates excellent generalization characteristics:

Observation: Test performance (F-macro=0.564) slightly exceeds development performance (0.557), with a difference of only +0.007. This pattern holds for the baseline as well (dev=0.518, test=0.520, +0.002).

Interpretation:

1. **No Overfitting to Dev Set**: The consistent dev/test alignment indicates that hyperparameter choices (C=500, TF-IDF settings) were not overfitted to development data. The improvements generalize robustly to unseen test data.
2. **Representative Splits**: The 90/10 stratified split successfully maintained the underlying data distribution. Both dev and test sets exhibit similar class imbalance and difficulty characteristics.
3. **Slightly Easier Test Set**: The marginal test superiority (+0.007) suggests the test set may contain slightly more discriminable cases or benefits from better class balance (test has proportionally more positive samples: 32.1% vs 32.0% in dev).

4. **Robust Features**: TF-IDF and bigrams prove to be stable features that do not exploit dev-specific artifacts. Unlike baseline's extreme training overfitting, the improved model's features transfer well across data partitions.

Confidence in Deployment: The consistent performance across splits (deviation ¡1%) suggests the improved model would maintain similar accuracy (±0.56) on new tweet data with comparable sentiment distribution.

5 Conclusion

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

IR Evaluation (Part 1): Implementing standard evaluation metrics from first principles reinforced understanding of precision-recall trade-offs and the importance of graded relevance (nDCG). The significance testing revealed that small sample sizes (n=10 queries) limit statistical power, emphasizing the need for large-scale evaluation collections. System 3's consistent superiority across multiple metrics (despite lack of statistical significance) demonstrates the value of multi-faceted evaluation.

Text Analysis (Part 2): The comparative analysis of MI, χ^2, and LDA illuminated complementary strengths: MI identifies exclusive vocabulary, χ^2 highlights high-frequency discriminators, and LDA reveals latent thematic structure. The striking differences in corpus coherence (Quran: 0.358, NT: 0.143, OT: 0.080) quantitatively capture the theological and compositional nature of religious texts—a finding that bridges computational methods with humanistic interpretation.

Text Classification (Part 3): The sentiment analysis task demonstrated the iterative nature of machine learning improvement. Error analysis directly informed feature engineering choices (bigrams for negation, TF-IDF for term weighting), resulting in substantial gains (+7.5% on dev, +8.5% on test). The consistent dev/test performance validated the robustness of improvements, while the largest gains on the negative class (+12%) confirmed that targeted error analysis yields actionable insights.

Methodological Lessons: Across all three parts, a common theme emerged: the necessity of critical evaluation. Statistical tests reveal when performance differences are meaningful (Part 1), feature selection methods must be evaluated for interpretability vs. discrimination (Part 2), and classification improvements must generalize beyond development data (Part 3). The coursework thus reinforced that computational text analysis requires not just implementing algorithms, but understanding their assumptions, limitations, and appropriate application contexts.

Challenges Faced: The most significant challenges included (1) correctly implementing nDCG with the specific lecture formula, (2) addressing Chi-square's negative correlation issue in feature selection, and (3) balancing model complexity to avoid overfitting in classification. Each challenge required consulting both theoretical foundations (lecture slides) and practical considerations (error analysis, cross-validation).

Future Directions: Potential extensions include: (1) larger query sets for IR evaluation to achieve statistical power, (2) hierarchical topic modeling (e.g., nested LDA) to capture subthemes within religious texts, and (3) transformer-based classifiers (e.g., BERT) for sentiment analysis to capture deeper contextual semantics beyond TF-IDF bigrams.