Information Retrieval and Text Mining

Modelling

Nuno Escudeiro (nfe@isep.ipp.pt)
Ricardo Almeida (ral@isep.ipp.pt)

Session outline

1. Information Retrieval modeling
2. Boolean Retrieval Model
    - Boolean algebra in IR
    - Disjunctive Normal Form (DNF) queries
    - Extensions of the Boolean Model

Learning outcomes

At the end of this session we will be able to:

- Explain what is an Information Retrieval (IR) model and describe its components
- Explain the Boolean Model for IR and describe its pros and cons
- Use Boolean Algebra in the context of IR
- Synthesize IR queries to Disjunctive Normal Form
- Discuss the applicability of Boolean IR for a given problem

1. IR Modeling

A general overview of document models for IR

IR Model

Modeling in IR is a complex process aimed at producing a ranking of documents based on their relevance to a given query.

An IR model involves two core elements:

- The conception of a logical framework for representing **documents** and **queries**
- The definition of a **ranking function** that computes a rank for each document regarding a given query

IR Model

- Modeling and Ranking
  - IR systems usually adopt index terms to index and retrieve documents
  - An index term is any word or expression that appears in the text of a document in the collection
  - Strengths
    - it can be implemented efficiently and it is simple to refer in a query. Simplicity reduces the effort of query formulation on the part of the user
  - Weaknesses
    - restricts the semantic of what can be expressed
    - abstracts document structure which can be of relevance
    - users have no training in properly forming their query

IR Model

- Modeling and Ranking
  - Central problem of an IR system: predicting which documents the user will find relevant and which ones are irrelevant
  - Solution: IR systems implement a predictive algorithm.

IR Model

- Characterization of an IR Model
  - Definition: An information retrieval model is a quadruple [**D, Q, F, R(q_i,d_j)**] where:
    - **D** is a set composed of logical views (or representations) of the documents in the corpus
    - **Q** is a set composed of logical views (or representations) of the users needs
    - **F** is a framework for modeling/representing documents, queries, and their relationships, such as:
      - — sets and boolean relations
      - — vectors and linear algebra operations
      - — sample spaces and probability distributions
      - — …
    - **R(q_i,d_j)** is a ranking function that associates a real number with a query representation **q_i ∈ Q** and a document representation **d_j ∈ D.**

IR Model

- Characterization of an IR Model
  - Definition: An information retrieval model is a quadruple [**D, Q, F, $R(q_i,d_j)$**] where:
    - **D** is a set composed of logical views (or representations) of the documents in the corpus
    - **Q** is a set composed of logical views (or representations) of the users needs
    - **F** is a framework for modeling/representing documents, queries, and their relationships, such as:
      - **sets and boolean relations**
      - vectors and linear algebra operations
      - sample spaces and probability distributions
      - …
    - **$R(q_i,d_j)$** is a ranking function that associates a real number with a query representation **$q_i \in Q$** and a document representation **$d_j \in D$.**

2. Boolean Model in IR

A general overview of the Boolean Retrieval Model: boolean algebra and basic operators

Boolean retrieval process

Boolean Retrieval Model

- The Boolean Retrieval Model is one of the earliest and simplest models used in information retrieval
- It is based on the Set Theory and Boolean Algebra (George Boole in the mid-19ths)
- Considers that index terms are present or absent in a document
- A query is essentially a conventional Boolean expression on index terms.

Boolean Model

- In the context of IR, the Boolean Model:
  - represents documents and queries as sets of terms or keywords, and
  - retrieval is performed using logical operations: AND, OR, and NOT.

Boolean Operators

- AND operator (∧):

A	B	A ∧ B
1	1	1
1	0	0
0	1	0
0	0	0

Boolean Operators

- OR operator (v):

A	B	A V B
1	1	1
1	0	1
0	1	1
0	0	0

Boolean Operators

- NOT operator (¬):

A	¬ A
1	0
0	1

Boolean Model

- Key characteristics and components of the Boolean model:
  - **Documents:** Each document in the collection is represented as a set of terms or keywords
  - **Queries:** rely on sets of terms or keywords
  - **Boolean Operators**:
    - **AND Operator (∧):** Retrieves documents that contain all the specified terms.
    - **OR Operator (∨):** Retrieves documents that contain at least one of the specified terms
    - **NOT Operator (¬):** Negates a term and retrieves documents not containing the specified term
  - **Boolean Expressions:** Queries are constructed by combining terms and operators into Boolean expressions
  - **Retrieval Process:** When a user submits a Boolean query, the retrieval system matches the query against the document collection using Boolean logic.

Boolean Model

- How does an IR system grounded on the Boolean framework ranks retrieved documents?

- **It does not!**
- In the Boolean Model of information retrieval, document ranking is not inherently performed. Instead, documents are retrieved in a binary manner – they are either relevant (match the query) or not relevant (do not match the query)
- Some extensions of the Boolean Model introduce ranking mechanisms.

Boolean Model

- **Basic Boolean Model (No Ranking)**
- **Query Representation**: The query is represented using Boolean operators (AND, OR, NOT)
- **Document Matching**: Documents are retrieved based on exact matches to the query conditions
- **Binary Relevance**: A document is either retrieved (1) or not (0); there is no ranking

- Query: **"machine AND learning"**
  - Document 1: *"Introduction to Machine Learning"* → ✅ (retrieved)
  - Document 2: *"Machine Vision and Deep Learning"* → ✅ (retrieved)
  - Document 3: *"Artificial Intelligence and Neural Networks"* → ❌ (not retrieved)
- Retrieved documents are not ranked; they are just returned as a set.

Boolean Model Extensions

- How does an IR system grounded on the Boolean framework ranks retrieved documents?

Ranking Extensions of the Boolean Model

- To introduce ranking in a Boolean retrieval system, heuristic methods can be applied.

Boolean Model Extensions

Term Frequency-Based Ranking

- Count the number of query terms present in each document.
- Documents containing more of the query terms are ranked higher.

For the query **"machine AND learning"**, if:

- Document 1 contains "machine" (5 times) and "learning" (3 times) → Score: **8**
- Document 2 contains "machine" (2 times) and "learning" (1 time) → Score: **3**
- **Ranking**:
- Document 1 (score 8) appears **before** Document 2 (score 3).

Boolean Model Extensions

Weighted Boolean Model (Term Importance)

- Assign different weights to query terms based on their importance (e.g., **TF-IDF** weighting).
- A document gets a higher score if it contains more **important** query terms.

- "AI" (is a common term, give it a low weight)
- "Quantum Computing" (is a rare term, give it a high weight).

Boolean Model Extensions

Proximity-Based Ranking

- If the query terms appear closer together in a document, it is ranked higher.

For the query **"machine AND learning"**

- Document A: "Machine learning is a subset of AI.“
- Document B: "Machine tools are useful, and learning is a continuous process.“
- Since "machine" and "learning" appear **closer** in Document A, it gets a higher rank.

Boolean Model Extensions

Document Length Normalization

- Shorter documents with exact query matches may be given **higher relevance** than long documents containing scattered matches

For the query **"machine AND learning"**

- Document A: "Machine learning is a subset of AI.“
- Document B: "Machine tools are useful, and learning is a continuous process.“
- Document A gets a higher rank.

Boolean Model Extensions

The **pure Boolean Model does not rank documents**, ranking can be introduced using several heuristics, such as:

- **Term Frequency-Based Ranking**
- **Weighted Boolean Model**
- **Proximity-Based Ranking**
- **Document Length Normalization**
- **…**

Boolean Model

Disjunctive Normal Form (DNF) is
a **sum of products**,
an **OR of AND terms**.

- All elements of the term-document matrix are either 1 (one), to indicate presence of the term in the document, or 0 (zero), to indicate absence of the term in the document
- A query *q* is a Boolean expression on the index terms such as, for instance, [q=ka ∧ (kb v ¬kc)]
- Given a query *q*, a term conjunctive component that satisfies its conditions is called a query conjunctive component c(q)
- By compiling all query conjunctive components, we can rewrite the query as a disjunction of those components. This is called the query **Disjunctive Normal Form**, which we refer to as q_{DNF}

Boolean Model

- In the Boolean model, a query is a conventional Boolean expression on index terms
  - Definition: Let c(q) be any of the query conjunctive components. Given a document d_j, let $c(d_j)$ be the corresponding document conjunctive component. Then, the similarity of the document d_j to the query q is defined as
  - $\text{sim}(d_j, q) = \begin{cases} 1 \; if \; \exists \; c(q) \mid c(\text{dj}) \\ 0 \; otherwise \end{cases}$

Boolean Model

- Term-document matrix
  - Is a fundamental data structure used to represent the relationship between terms and documents in a document collection where:
    - rows represent terms (or words)
    - columns represent documents
    - each cell of the matrix contains a numerical value that indicates the frequency or some other measure of the presence of the corresponding term in the respective document.
    - In the case of the Boolean model the value is true (1) or false (0)

Boolean Model

- **Term-document matrix**:
  - Let us consider transportation vehicles and the following document collection:
    1. Car: "A car is a common mode of transportation.“
    2. Train: "Trains are used for long-distance travel."
    3. Plane: "Planes are fast and efficient for air travel.“
    4. Boat: "Boats are essential for water transportation.“
  - Also consider the following terms:
    - car, train, plane, boat, transportation, travel, mode, fast, efficient, water, air, long-distance

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	1	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	1	0
Mode	1	0	0	0
Fast	0	0	1	0

Q1 = Car OR Boat

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	1	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	1	0
Mode	1	0	0	0
Fast	0	0	1	0

Q1 = **Car** OR **Boat**

{**Doc1**, **Doc4**}

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	1	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	1	0
Mode	1	0	0	0
Fast	0	0	1	0

Q1 = **Car** OR **Boat**

{**Doc1**, **Doc4**}

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	1	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	1	0
Mode	1	0	0	0
Fast	0	0	1	0

Q1 = Car OR Boat

Q2 = (Travel AND Fast) OR (Train)

{**Doc2**, **Doc3**}

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	**1**	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	**1**	0
Mode	1	0	0	0
Fast	0	0	**1**	0

Q1 = Car OR Boat

Q2 = (Travel AND Fast) OR (Train)

{**Doc2**, **Doc3**}

Boolean Model

- Term-document matrix:

	Doc1	Doc2	Doc3	Doc4
Car	1	0	0	0
Train	0	1	0	0
Plane	0	0	1	0
Boat	0	0	0	1
Transportation	1	0	0	1
Travel	0	1	1	0
Mode	1	0	0	0
Fast	0	0	1	0

Q1 = Car OR Boat

Q2 = (Travel AND Fast) OR (Train)

DNF

Disjunctive Normal Form

In the **Boolean Model** of Information Retrieval, the **Disjunctive Normal Form (DNF)** of a query is obtained by rewriting the Boolean expression as a **disjunction (OR) of conjunctions (ANDs)** of terms.

Steps to Obtain Disjunctive Normal Form (DNF)

1. **Expand the Query Using Boolean Laws**
   - Apply **distribution** of AND over OR to rewrite the expression.
   - Use **De Morgan's Laws** to simplify NOT operations when necessary.
2. **Express the Query as a Disjunction of Conjunctions**
   - Each **conjunctive clause** (AND clause) represents a set of terms that must be present together in a document.
   - The **OR** operator combines multiple conjunctive clauses.

Disjunctive Normal Form

Disjunctive Normal Form (DNF) is
a **sum of products**,
an **OR of AND terms**

In the **Boo**… **F)** of a
query is o…
conjunctio…

De Morgan's Laws

1st Law (Negation of AND):

¬(A ∧ B) = ¬A ∨ ¬B

•The negation of a conjunction is the disjunction of the negations.

"Not (A and B)" is the same as "Not A or Not B."

2nd Law (Negation of OR):

¬(A ∨ B) = ¬A ∧ ¬B

•The negation of a disjunction is the conjunction of the negations.

•"Not (A or B)" is the same as "Not A and Not B."

If **A = "It is raining"** and **B = "It is cold"**

Then **"It is not raining and it is not cold"** is the same as **"It is not (raining or cold)."**

Steps to O…

1. **Expand**…
   - App…
   - Use…
2. **Express**…
   - Each **conjunctive clause** (AND claus… together in a document.
   - The **OR** operator combines multiple…

Why a
Disjunctive Normal Form (OR of AND terms)
… and not a
Conjunctive Normal Form (AND of OR terms)?

Boolean Laws

1. Identity Laws
- $A \land 1 = A$
- $A \lor 0 = A$

(AND with 1 or OR with 0 does not change A)

2. Null Laws (Dominance Laws)
- $A \land 0 = 0$
- $A \lor 1 = 1$

(AND with 0 results in 0, OR with 1 results in 1)

3. Idempotent Laws
- $A \land A = A$
- $A \lor A = A$

(AND or OR with itself does nothing)

4. Complement Laws
- $A \land \neg A = 0$
- $A \lor \neg A = 1$

(A AND NOT A is always false, A OR NOT A is always true)

5. Double Negation Law
- $\neg(\neg A) = A$

(Negation cancels out)

6. Commutative Laws
- $A \land B = B \land A$
- $A \lor B = B \lor A$

(Order doesn’t matter for AND/OR)

7. Associative Laws
- $(A \land B) \land C = A \land (B \land C)$
- $(A \lor B) \lor C = A \lor (B \lor C)$

Grouping doesn’t matter for AND/OR)

8. Distributive Laws
- $A \land (B \lor C) = (A \land B) \lor (A \land C)$
- $A \lor (B \land C) = (A \lor B) \land (A \lor C)$

(AND distributes over OR, OR distributes over AND)

9. Absorption Laws
- $A \lor (A \land B) = A$
- $A \land (A \lor B)=A$

(Simplifies complex expressions)

10. De Morgan’s Laws
- $\neg(A \land B) = \neg A \lor \neg B$
- $\neg(A \lor B)=\neg A \land \neg B$

(Negation flips AND/OR)

Disjunctive Normal Form

Why Use DNF in Information Retrieval?

- It **simplifies Boolean queries** for retrieval engines.
- Each **conjunctive clause (AND clause)** represents a **minimal document retrieval condition**.
- Allows **efficient document filtering** by breaking down queries into simpler conditions.

Disjunctive Normal Form

Example 1: Basic Boolean Query

Given query:

$(A \wedge (B \vee C))$

Step-by-step transformation:

1. Distribute **AND** over **OR**:

$(A \wedge B) \vee (A \wedge C)$

2. The result is now in **DNF**:

$(A \wedge B) \vee (A \wedge C)$

- This means a document must contain **(A and B) OR (A and C)** to be retrieved.

Disjunctive Normal Form

Example 2: More Complex Query

Given query:

$(A \lor B) \land (C \lor D)$

Step-by-step transformation:

1. Apply distribution:

$(A \land C) \lor (A \land D) \lor (B \land C) \lor (B \land D)$

2. This is now in **DNF** form:

$(A \land C) \lor (A \land D) \lor (B \land C) \lor (B \land D)$

- This means a document must contain **any of the AND-combinations** to be retrieved.

Applying DNF to document retrieval

Scenario: Searching a Document Collection

Assume we have a **Boolean query** for retrieving documents from a collection of articles.

User Query:

(Machine ∨ Deep) ∧ (Learning ∨ AI)

- This means the user wants documents containing **either "Machine" or "Deep"**, and at the same time, containing **either "Learning" or "AI"**.

Applying DNF to document retrieval

Step 1: Convert to DNF

- Using **distribution** of AND over OR:

$(\text{Machine} \land \text{Learning}) \lor (\text{Machine} \land \text{AI}) \lor (\text{Deep} \land \text{Learning}) \lor (\text{Deep} \land \text{AI})$

- Now, the query consists of four conjunctive (AND) clauses:
  - **Machine AND Learning**
  - **Machine AND AI**
  - **Deep AND Learning**
  - **Deep AND AI**

Applying DNF to document retrieval

Step 2: Apply to a Document Collection

- Assume we have the following **documents** in our system:

D1 - "Machine Learning is a key area of AI research."

D2 - "Deep Learning is revolutionizing AI applications."

D3 - "Machine intelligence is different from Large Language Models."

D4 - "AI and Machine Learning work together in modern computing."

D5 - "Deep Learning and AI are driving innovations."

Applying DNF to document retrieval

Step 3: Retrieve Matching Documents

Now, we check which documents satisfy **at least one** of the four DNF clauses:

1. **(Machine AND Learning)** → ✅ Matches **D1, D4**
2. **(Machine AND AI)** → ✅ Matches **D1, D4**
3. **(Deep AND Learning)** → ✅ Matches **D2, D5**
4. **(Deep AND AI)** → ✅ Matches **D2, D5**

Thus, the retrieved documents are:
D1, D2, D4, D5 (D3 is excluded since it doesn’t match any clause).

Applying DNF to document retrieval

Corpus

D1 - "**Machine Learning** is a key area of AI research."

D2 - "Deep Learning is revolutionizing AI applications."

D3 - "Machine intelligence is different from Large Language Models."

D4 - "AI and **Machine Learning** work together in modern computing."

D5 - "Deep Learning and AI are driving innovations.“

Query

(Machine AND Learning) OR (Machine AND AI) OR (Deep AND Learning) OR (Deep AND AI)

Applying DNF to document retrieval

Corpus

D1 - "**Machine Learning** is a key area of **AI** research."

D2 - "Deep Learning is revolutionizing AI applications."

D3 - "Machine intelligence is different from Large Language Models."

D4 - "**AI** and **Machine Learning** work together in modern computing."

D5 - "Deep Learning and AI are driving innovations.“

Query

(Machine AND Learning) OR (Machine AND AI) OR (Deep AND Learning) OR (Deep AND AI)

Applying DNF to document retrieval

Corpus

D1 - "**Machine** **Learning** is a key area of **AI** research."

D2 - "**Deep Learning** is revolutionizing AI applications."

D3 - "Machine intelligence is different from Large Language Models."

D4 - "**AI** and **Machine** **Learning** work together in modern computing."

D5 - "**Deep Learning** and AI are driving innovations.“

Query

(Machine AND Learning) OR (Machine AND AI) OR (Deep AND Learning) OR (Deep AND AI)

Applying DNF to document retrieval

Corpus

D1 - "**Machine Learning** is a key area of **AI** research."

D2 - "**Deep Learning** is revolutionizing **AI** applications."

D3 - "Machine intelligence is different from Large Language Models."

D4 - "**AI** and **Machine Learning** work together in modern computing."

D5 - "**Deep Learning** and **AI** are driving innovations.“

Query

(Machine AND Learning) OR (Machine AND AI) OR (Deep AND Learning) OR (Deep AND AI)

Applying DNF to document retrieval

Corpus

D1 - "**Machine Learning** is a key area of **AI** research."

D2 - "**Deep Learning** is revolutionizing **AI** applications."

~~D3 - "Machine intelligence is different from Large Language Models."~~

D4 - "**AI** and **Machine Learning** work together in modern computing."

D5 - "**Deep Learning** and **AI** are driving innovations.“

Query

(Machine AND Learning) OR (Machine AND AI) OR (Deep AND Learning) OR (Deep AND AI)

Applying DNF to document retrieval

Step 4: Ranking (Optional)

Since the **Boolean Model** doesn’t inherently rank results, we can use:

- **Term Frequency (TF)**: Rank documents higher if they contain the query terms more frequently.
- **Proximity Ranking**: Rank documents higher if the matched words appear closer together.

For example:

- **D1** could be ranked higher than **D4**, both have “Machine” and “Learning" close by but in D1 it appears first.

Applying DNF to document retrieval

Step 4: Ranking (Optional)

Final Retrieved Results (Ranked)

1. **D5** – "Deep Learning and AI are driving innovations."
2. **D2** – "Deep Learning is revolutionizing AI applications."
3. **D1** – "Machine Learning is a key area of AI research."
4. **D4** – "AI and Machine Learning work together in modern computing."

Boolean Model

Highlights:

- **DNF breaks complex queries into smaller, manageable conditions.**
- **Documents are retrieved based on logical AND conditions.**
- **Ranking can be added for better result ordering.**

Boolean Retrieval helps in **efficiently filtering** documents before applying more advanced ranking techniques.

Boolean Model

- Term-document matrix:
  - Let us consider an automobile and the following corpus:
    1. Doc1: "An automobile consists of various components, including wheels and a spark plug."
    2. Doc2: "Regular maintenance of automobiles involves checking the condition of wheels and replacing the spark plug."
    3. Doc3: "The invention of the spark plug revolutionized automobile engines, leading to improved performance."
    4. Doc4: "Wheels are an essential part of automobiles, providing mobility, while spark plugs ignite fuel in the engine."
  - Also consider the following terms / vocabulary / lexicon:
    - {automobile, wheels, spark plug}
  - And query q
    - q = (automobile ∧ wheels) v spark plug

Boolean Model

- Term-document matrix:
  - Let us consider an automobile and the following corpus:
    1. **Doc1: "An automobile consists of various components, including wheels and a spark plug."**
    2. **Doc2: "Regular maintenance of automobiles involves checking the condition of wheels and replacing the spark plug."**
    3. **Doc3: "The invention of the spark plug revolutionized automobile engines, leading to improved performance."**
    4. **Doc4: "Wheels are an essential part of automobiles, providing mobility, while spark plugs ignite fuel in the engine."**
  - Also consider the following terms / vocabulary / lexicon:
    - {automobile, wheels, spark plug}
  - And query q
    - q = (automobile ∧ wheels) v spark plug)

Boolean Model

- Term-document matrix:
  - Let us consider an automobile and the following corpus:
    1. Doc1: "An automobile consists of various components, including wheels and a spark plug."
    2. Doc2: "Regular maintenance of automobiles involves checking the condition of wheels and replacing the spark plug."
    3. Doc3: "The invention of the spark plug revolutionized automobile engines, leading to improved performance."
    4. Doc4: "Wheels are an essential part of automobiles, providing mobility, while spark plugs ignite fuel in the engine."
  - Also consider the following terms / vocabulary / lexicon:
    - **{automobile, wheels, spark plug}**
  - And query q
    - q = (automobile ∧ wheels) v spark plug)

Boolean Model

- Term-document matrix:
  - Let us consider an automobile and the following corpus:
    1. Doc1: "An automobile consists of various components, including wheels and a spark plug."
    2. Doc2: "Regular maintenance of automobiles involves checking the condition of wheels and replacing the spark plug."
    3. Doc3: "The invention of the spark plug revolutionized automobile engines, leading to improved performance."
    4. Doc4: "Wheels are an essential part of automobiles, providing mobility, while spark plugs ignite fuel in the engine."
  - Also consider the following terms / vocabulary / lexicon:
    - {automobile, wheels, spark plug}
  - And query q
    - **q = (automobile ∧ wheels) ∨ spark plug)**

Boolean Model

- Term-document matrix:

q = (automobile ∧ wheels) ∨ spark plug

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

Ranking?

D1, D2, D3, D4

Boolean Model

- Term-document matrix:

q = (automobile ∧ wheels) v spark plug

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

D1, D2, D4, D3

Boolean Model

- Term-document matrix:

DNF?

q = (automobile v wheels) ʌ (wheels v spark plug)

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

Boolean Model

- Term-document matrix:

~~q = (automobile ∨ wheels) ∧ (wheels ∨ spark plug)~~

q = (automobile ∧ spark plug) ∨ wheels

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

Ranking?

D1, D2, D4, D3

Boolean Model

- Term-document matrix:

DNF?

q = (automobile ∧ wheels) ∨ ¬(wheels ∧ spark plug)

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

Boolean Model

- Term-document matrix:

~~q = (automobile ∧ wheels) ∨ ¬(wheels ∧ spark plug)~~

q = automobile V ¬wheels V ¬spark

	Doc1	Doc2	Doc3	Doc4
automobile	1	1	1	1
wheels	1	1	0	1
spark plug	1	1	1	1

Ranking?

D3, D1, D2, D4

Boolean Model

- Limitations:
  - Since the decision is binary, without any notion of grading scale, it limits retrieval quality
  - It is not easy to translate information needs into boolean expressions

References

- https://users.dcc.uchile.cl/~rbaeza/mir2ed/
- https://machinelearningmastery.com/
- https://www.evidentlyai.com/classification-metrics/explain-roc-curve