Fusion: Introduction

COMP3009J: Information Retrieval

Dr. David Lillis (david.lillis@ucd.ie)

UCD School of Computer Science
Beijing Dublin International College

Introduction

- You have already seen that there are many algorithms that can be used for Information Retrieval (IR):
  - Boolean Model
  - Vector Space Model
  - Probabilistic Model
  - BM25
  - … many more
- If there was one algorithm that worked better than all the others all the time, we could just use that.
- There isn't!

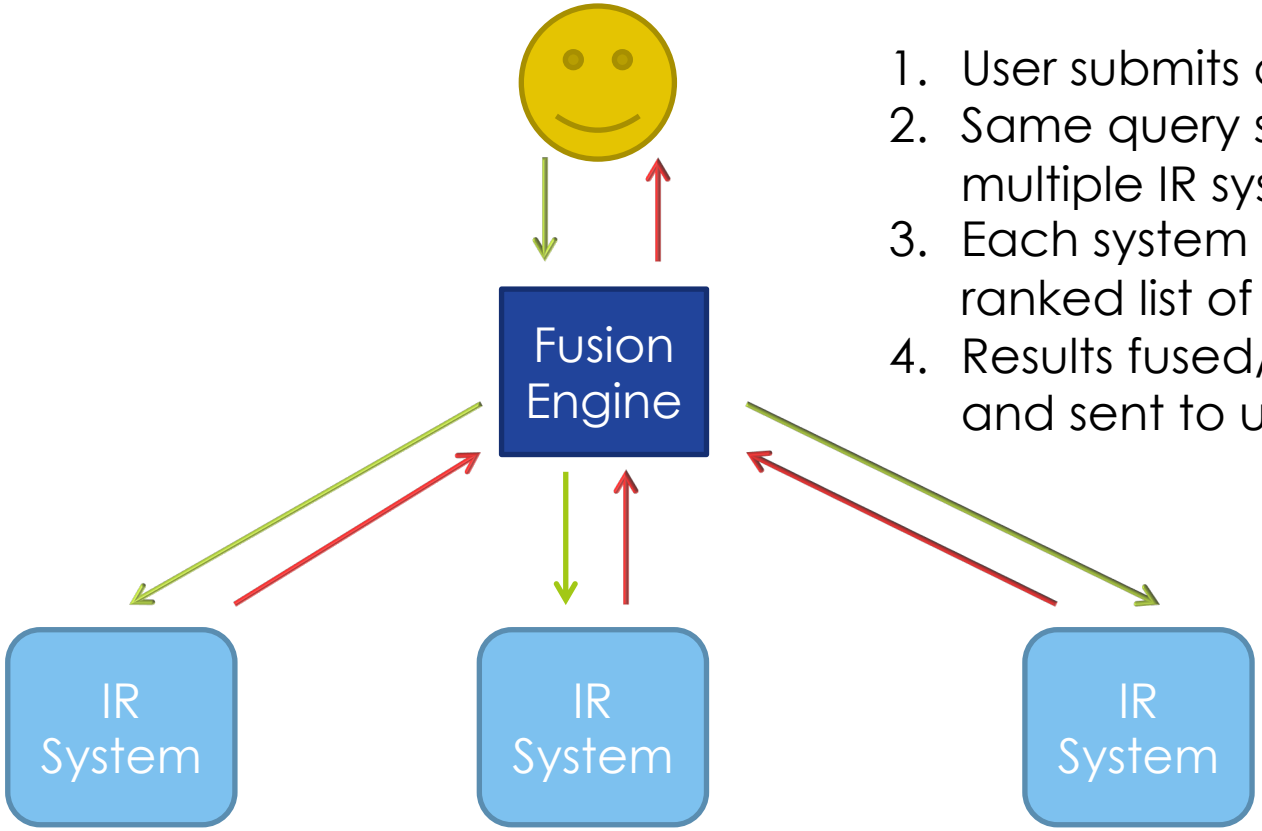
Introduction

- From the 1990s, an area of research called ***data fusion***, ***collection fusion***, or ***results aggregation*** became increasingly popular.
- **Basic definition:**
  - **Combining** the outputs of **multiple** different Information Retrieval systems/algorithms into a single ranked result set that can be shown to a user in response to a **query**.
- It is hoped that this combined result set will be of **better quality** than the invidual input result sets.
  - Higher precision, higher recall (and other metrics).

Better Quality?

- But what do we mean by *better quality*?
  - Higher **Recall** – almost inevitable if we bring in an additional set of results (more of the available relevant documents are retrieved)
  - Higher **Precision** – more difficult. Need to make sure we introduce relevant documents in place of non-relevant documents.
    - For metrics (such as MAP) that reward high positions for relevant documents, we should ensure that our ranking puts more relevant documents first.

Anatomy of a Fusion System



1. User submits query
2. Same query sent to multiple IR systems
3. Each system returns a ranked list of results
4. Results fused/merged and sent to user.

Anatomy of a Fusion System

- The final list returned to the user is typically ranked by calculating a **score** for **each document**: the document with the highest score goes at the beginning.
  - Document contents are not considered.
- Generally, each underlying IR system will contribute something towards the score of each document it returns.
  - Might depend on the **position/rank** of the document in the result, the **score** explicitly given by the search engine, **quality** of the search engine itself.
- Main difference between algorithms: what information they take into account when allocating scores.

Applications of Fusion

- **Metasearch**: This involves fusion of result sets returned by **autonomous, complete search engines** (e.g. Google, Bing).
- **Distributed Information Retrieval:** Numerous IR systems are **designed to co-operate** with one another, each working on a subset of the document collection.
- **Internal Metasearch:** Numerous algorithms perform searches on the **same document collection** (data fusion).

Corpus Overlap

- Before developing a fusion algorithm, it is important to consider how much the document collections (corpora) used by the systems we wish to use **overlap**.
- There are three different **levels of overlap** that can occur between document corpora.
- The level of overlap will have a significant effect on how we treat the result sets when fusing.

Corpus Overlap: Disjoint Databases

- Here, the input systems search separate, disjoint document collections that have **no documents in common**.
- A document **cannot be returned by more than one input system**, since it does not appear in more than one index.
- Distributed IR is typically implemented using disjoint databases.
- Fusion of disjoint corpora is frequently known as *Collection Fusion*.

Corpus Overlap: Identical Databases

- The input systems each apply their own IR algorithm to the **same set of documents**.
- Documents will frequently appear in multiple result sets.
- Appearing in multiple result sets is frequently interpreted as further **evidence of relevance** (as multiple systems agree that it is relevant).
- This has become known as ***Data Fusion*** and is our **main focus**.
- Internal Metasearch is generally a Data Fusion task.

Corpus Overlap: Overlapping Databases

- The document collections being used by the various input systems have some level of overlap, but are **not identical**.
- Documents may appear in multiple result sets.
- However, it is **difficult to draw reliable conclusions** about these documents that appear in multiple result sets.
- External Metasearch generally involves overlapping databases.

Corpus Overlap: Overlapping Databases

- A common feature of fusion algorithms is to give a **higher score** to documents appearing in **multiple result sets**:
  - Appears in every result set => every system considers it relevant.
  - Appears in no result sets => no system considers it relevant.
  - Appears in one result set but not another =>
    - One system does not consider it to be relevant **OR**
    - One system is not aware of the document.
- Difficult to decide how to treat the last situation, as it's often impossible to tell which possibility has occurred.

Three Fusion “Effects”

The Skimming Effect

- There are three "**effects**" that a fusion algorithm may try to leverage*
- In general, the most relevant documents in a result set appear at or near the top, when an effective IR algorithm is being used.
- The Skimming Effect argues that **"skimming" the top documents** from each result set and using these for fusion should give better performance.
- This principle is used for **all popular fusion algorithms**.

* Vogt, C. C., & Cottrell, G. W. (1999). Fusion Via a Linear Combination of Scores. *Information Retrieval*, *1*(3), 151–173

The Chorus Effect

- If **multiple input systems agree** that a document is relevant, the Chorus Effect argues that this evidence of relevance should be taken into account and that document should be **highly ranked** in the fused result set.
- Whether this effect is applicable depends on the level of overlap between the corpora used by the input systems.
- For Data Fusion, the Chorus Effect tends to be an important consideration.
- For Collection Fusion, it is not a factor at all (documents cannot appear in multiple result sets).
- Difficult to gauge for partially overlapping corpora.

The Chorus Effect

Document d_5 is given a high rank by both systems: should probably be ranked highly in fused result set (Skimming & Chorus Effects)

System A	
Rank	**Document**
1	d_{19}
2	d_5
3	d_{12}
4	d_4
5	d_{14}
6	d_{15}
7	d_1
8	d_9
9	d_{10}
10	d_{11}

System B	
Rank	**Document**
1	d_5
2	d_{14}
3	d_{20}
4	d_7
5	d_1
6	d_{11}
7	d_{18}
8	d_3
9	d_{10}
10	d_{12}

The Chorus Effect

System A	
Rank	**Document**
1	d_{19}
2	d_5
3	d_{12}
4	d_4
5	d_{14}
6	d_{15}
7	d_1
8	d_9
9	d_{10}
10	d_{11}

Document d_{19} is given a high rank by one system but is not returned at all by System B: should be ranked below d_5 according to the Chorus Effect.

System B	
Rank	**Document**
1	d_5
2	d_{14}
3	d_{20}
4	d_7
5	d_1
6	d_{11}
7	d_{18}
8	d_3
9	d_{10}
10	d_{12}

The Dark Horse Effect

- The Dark Horse Effect is where one input system returns results of a much **different quality** than the others.
- This may be as a result of returning either **unusually accurate** or **unusually inaccurate** results.
- This seems to **contradict the Chorus Effect**, as it would favour identifying a "dark horse" and just using its results, rather than fusing it with others.
- The Dark Horse Effect is very difficult to identify, and so is generally not used in fusion algorithms.

Categories of Data Fusion Techniques

- There are three principal categories of Data Fusion techniques that we will look at:
  - **Rank-Based Fusion**: These examine only the **rank/position** that each document occupies in the input result sets. Sometimes necessary if relevance scores are unavailable.
    - **Voting models** generally operate on the rank level also.
  - **Score-Based Fusion**: The **relevance scores** given to a document by an input system can be used as a measurement of how confident it is as to the document's relevance.
  - **Segment-Based Fusion**: Result sets are divided into groups of documents, rather than using individual ranks or scores.