Behaviour of tf-idf and how it is used in practice

Question:

Why, if we have 2 keywords given and we find one document with a very high score for one keyword, but it doesn't appear in the second keyword's documents, do we discard that document? If we use the formula on page 17, workshop4.pdf) it should give that document. What if I typed "department arachnocentric", and only one document x has a word "arachnocentric", but x doesn't have a word "department", I think we should get that document, but we wouldn't as you said in the lecture.

Response:

Well spotted, but as usual the answer is: it depends. E.g.,

- You can require each term to be present.
- For any term not present, you can set its tf-idf score to 0.

It comes down to saying: is a document that mentions only term X and not term Y of greater use than one that mentions both (or: if I ranked a bunch of documents containing only term X above those containing both, would I be giving the user a false impression that there might be no document about both, especially if the documents with both didn't appear until a couple of result pages later)?

By the way, the terms with 0 in the flist?.weights files had that score essentially because they were found in many/all documents at a high rate, not because they weren't there, e.g., words like 'a', 'and', 'but', 'by', ..., so you could be talking about 2 different 0s: 0 = 'uninformative' and 0 = 'not there'. But then you could arguably think of these as essentially the same.

I was trying to make things easier in the exercise by telling you to require presence of all terms.

Apache Lucene, which is a popular 'industrial strength' IR package, has this to say in relation to tf-idf:

"A document may match a multi term query without containing all the terms of that query, and users *[i.e., developers implementing a Lucene instance]* can further reward documents matching more query terms through a coordination factor, which is usually larger when more terms are matched: coord-factor(q,d)."

See also the IIR book, section 7.3, where they say:

"Classically, the interpretation of free text queries was that at least one of the query terms be present in any retrieved document. However, more recently, web search engines have popularized the notion that a set of terms typed in carries the semantics of a conjunctive query that only retrieves documents containing all or most query terms."

So, in practice, you'd normally want to impose a factor à la Lucene to avoid situations like the one I described above where you could mislead a user without them realising (until they actually read a top ranked document to find that it did not cover all the concepts they were interested in).

In other words, there's typically a difference between the pure theory and the normal IR practice.