Finding Information

What does it cost, what does it take?

imprima | White Paper

Published April 2021

By Pieter van der Made (Executive Chairman) and Jeroen de Knijf (Senior Data Scientist) at Imprima

The contents of this document are limited to general information and not detailed analysis of law or legal advice and are not intended to address specific legal queries arising in any particular set of circumstances.

Introduction

Do you feel you are wasting a lot of time looking for information and documentation? Well, you are not alone. According to McKinsey, 'employees spend 1.8 hours every day – 9.3 hours per week, on average – searching and gathering information' [1]. IDC data shows that 'the knowledge worker spends about 2.5 hours per day, or roughly **30% of the workday, searching for information**'.

These numbers are quite shocking. But it is indeed reality, even more so in M&A Due Diligence, where the objective is, first and foremost, just to find information. And that can be challenging. For example, finding red flags in a large number of agreements is often like looking for a needle in a haystack. Imagine how much productivity would improve if information could be found more easily. In addition, again in particular in Due Diligence, time is limited, so therefore the likelihood of missing certain information is high, reducing the quality of Due Diligence.

So how do we search for information in documentation? **First of all, we must find the relevant documents. Only then can we find the information IN the documents.**

Traditional Search Methods

Searching for documents in a data room, certain locations or folders

In case of a Due Diligence project, the data is normally stored in a Virtual Data Room, where the documents are organised in a hierarchical folder structure. Due Diligence professionals (lawyers at the advising law firm or analysts at investment banks) will try to find documents on the basis of this structure (In non-Due Diligence situations the problem is not much different: in most organisations, documents are stored in hierarchical structures). **The problem is that the location where documents are saved in this structure is ambiguous and subjective**. Different people make different choices as to where to save documents. Unless the information you are looking for maps 1:1 of this structure, it is often challenging to find the relevant documentation, and hence the information in it.

Tagging documents with keywords

An alternative is to tag documents with certain keywords, so that a search can be done on the keywords of the documents. However, again, **the way documents are tagged is ambiguous and subjective**. You are very dependent on how others have tagged

the documents, and whether the tags are relevant for your query, and whether they have been tagged at all.

Full text search and its limitations

Yet another alternative is to search the whole set of documents, and all their content, with a 'full text search' (and some logic captured in so called 'regular expressions'): trying to find the documents that you need by looking for those documents that contain certain terms.

Though this is very flexible, full text search is not very intuitive, and does not capture the semantics of what you are looking for. To use an everyday example: if you google "vegetable" "garden" and "rabbits", you are probably not looking for documents that contain the words "vegetable" "garden" and "rabbits": you are more likely to instead be looking for documents or tips that give advice how to keep rabbits out of your vegetable garden (for instance). But a full text search will not accomplish the latter.

As a result, **a full text search is not very accurate, and often even not very useful**. It is likely to return many documents that contain the keyword you entered that are actually not relevant for you (in other words, low 'Recall' (for more information on 'Recall' and why it matters, see our previous whitepaper 'AI that really works'). That means you will still spend a lot of time ploughing through the documents only to determine if they are relevant at all. At the same time, what such a search returns is likely not to include all documentation you are looking for (in other words, low 'Precision', again, see 'AI that really works').

Download your FREE COPY of this whitepaper

Machine Learning-Based Search

What we need is something that is:

- **as flexible as a keyword search;**
- **allows you to search for anything;**
- **and at the same time has a high recall (so you don't miss anything) at an acceptable precision (so you don't have to plough through any irrelevant docs before finding the docs you need).**

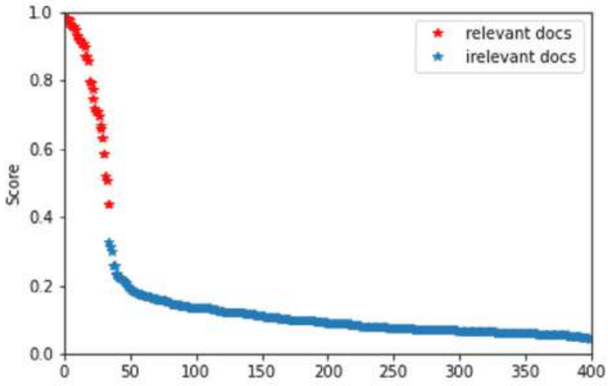
So how do we search for information in data that is neither structured according to our needs nor is tagged with appropriate keywords? (spoiler alert: it never is). Answer: ignore the structure, don't use keywords, and overcome the limitations of full text search by using **a comprehensive measure of the content and context of the entire text in the documents**. The challenge is then to have an algorithm that is able to determine, as a human would, from the entirety of the text what is relevant for your query. Sounds complicated? That is where Machine Learning (ML) comes in: The machine learning, as a matter of speech, observes how you search for the information and what is important to you, it learns to mimic your way of working.

Case study: Finding Libor Clauses

Let's not go into how that technically works, but show an example now.

Objective here is to find LIBOR-related clauses in credit agreements. And we don't just want to find those clauses, we want to find the documents that contain them. More specifically, we want to find credit agreements that contain clauses referring to legislation changes or other contingencies that would the invalidate the usage of LIBOR as a benchmark for interest calculation. Note that that is much more specific than finding credit agreements that contain the word LIBOR, which would give us many documents that would not be relevant for us. Our test data set consists of circa 400 docs. We now want the AI to find the documents that contain such concepts, by assigning a higher score to the documents that are most likely to be relevant (i.e. containing these concepts) and placing them at the top of the list.

The results are shown below. As a result of the 'score' the ML assigns to each document, the relevant ones (the red dots) are all listed before the non-relevant ones (the blue dots). So as a user, you would not have to plough through all 400 documents to find the relevant ones. In this case, you only need to review the first 30 documents with the highest-relevancy score to find all the relevant ones. Clearly, this saves a tremendous amount of time.



Picture 1. You only need to review the first 30 documents with the highest-relevancy score to find all the relevant documents from the dataset of 400 documents.

Data Vs Process

As you can see from the above example, ML can be very successful to find the clauses. Moreover, this was achieved by supplying only one example clause to the algorithm, and then just accepting the relevant documents. That is enough to get the ML fully trained. Let's discuss how that is achieved.

Most AI technologies rely heavily on existing data. However, in practice that data is often not available. In Due Diligence, a law firm cannot use one customer's data to train ML algorithms to subsequently predict in the dataset of another customer, at least not just like that. And even if that is possible, 'Transfer learning', i.e. how to adapt the ML algorithm trained on one data set to be effective and accurate on another, is a big issue. As a matter of fact, and paradoxically, the more you rely on ML trained on external data, the more difficult it will be to train the ML for your own purposes on your own data.

Instead, at Imprima, our approach is to enable a process that allows you to use the dataset at hand, and only that data. Training on your own data does not only enable the ML to learn very fast, but it also allows it to be **truly language independent**.

So what, one might say, as ML being language agnostic is a much-claimed feature. As a matter of fact, any ML algorithm is, in principle, language independent, so that in itself is nothing special. **However, that will not be the case anymore once trained in a certain language, which would happen if your ML relied on external data.**

And it is not just language. **What about jurisdiction**? An agreement in English under Dutch law will read very differently than an agreement in English under English law. The ML will have to be trained differently as well, accordingly. The same principle of using your own data and your own behaviour to train the ML, also solves the jurisdiction 'problem'.

That's why, at Imprima we enable the ML process, without relying on external data.

To see how ML is used in a real-life due diligence example, please visit our Smart Review product page or contact us to learn more.

www.imprima.com

Contact us today to find out more

London
Tel: +44 20 7965 4700
E: londonsales@imprima.com

Paris
Tel: +33 1 58 36 06 60
E: paris@imprima.com

Milan
Tel: +39 0694 803 478
E: milan@imprima.com

Frankfurt
Tel: +49 699 150 9800
E: frankfurt@imprima.com

Amsterdam
Tel: +31 207 155 600
E: amsterdam@imprima.com

Brussels
Tel: +33 1 76 75 32 91
E: brussels@imprima.com