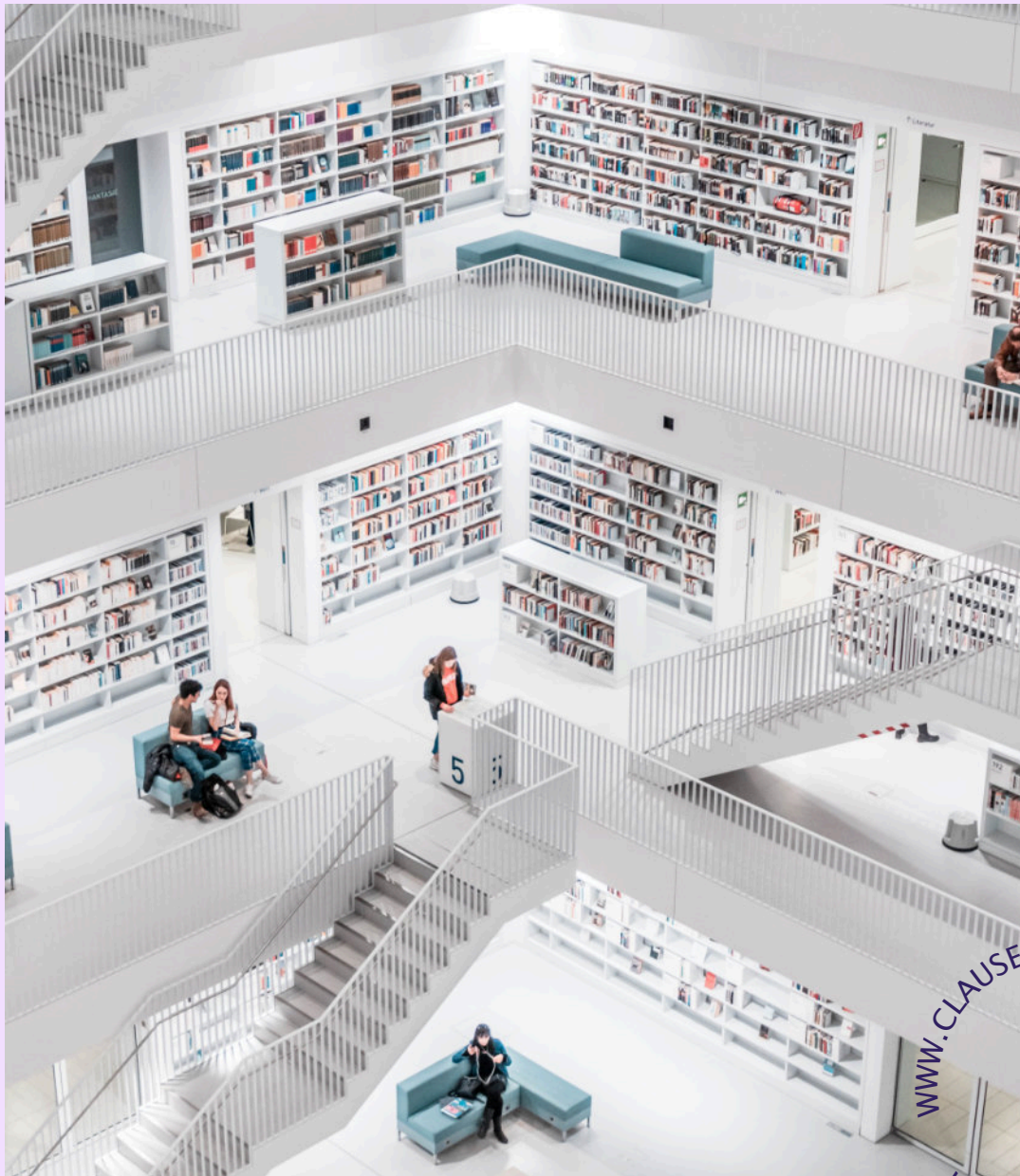


The Ultimate Guide to Building Clause Libraries.



INTRODUCTION	1
CHAPTER 01 What's the problem?	2
Story time: Lucille drafts a license agreement	3
Nuance 1: "Our client is the licensor"	4
Nuance 2: "The subject matter is a trademark license"	5
Nuance 3: "The license should be non-sublicensable and remunerated"	5
CHAPTER 02 How can clause libraries help?	7
Speed	8
Education	8
Compliance	9
Mitigating the risks of precedent-based drafting	9
Consistency	9
Language versions	10
Access control	10
CHAPTER 03 What tools can I use to build clause libraries?	11
The "do it yourself" single document approach`	13
Dedicated clause library tools	18
Document management and document automation/assembly tools	20
CHAPTER 04 How do I start building a clause library?	21
Question 1: Who needs access?	22
Question 2: How to set up my folder structure?	22
Option 1 - Based on legal domain	22

Option 2 - Based on the document type	23
Question 3: Which clauses do I add to my clause library?	24
Question 4: How should I augment my clauses?	27
Question 5: Who do you need to build a clause library?	28
BONUS question: How time-consuming is building a clause library?	28
CHAPTER 05 The role of AI in building clause libraries	29
Keyword-focus	30
BONUS ROUND: How does AI work?	32
You are all set!	36

INTRODUCTION

When it comes down to it, legal work is all about sharing knowledge. Legal experts arm themselves with knowledge of the law.

Sure, there are transactional lawyers and litigation lawyers; corporate lawyers and employment lawyers; in-house counsel and attorneys. But whatever their place in the legal services sector, lawyers apply their knowledge to solve problems, and they typically do so by providing it in a document format (e.g.: a contract, a memo, an email, a petition to the court, etc.)

With this in mind, the importance of being able to make that knowledge available in the most user-friendly, most efficient way possible is what sets otherwise equal legal experts apart.

But while a lot of time and effort goes into creating documents, strikingly little thought goes into how to best capture the knowledge within for future projects to boost the internal workings of the organisation. This leads to a constant reinventing of the wheel, as legal experts individually do drafting work they have done previously and new team members do drafting work their seniors have done before them.

Enter: the clause library

Clause libraries (also known as a “clause bank” or “precedent library”) are repositories filled with standardised template clauses that have been pre-constructed and pre-approved, and ideally give legal experts a one-stop shop for their drafting needs.

Many different approaches to the clause library exist, and we will be looking at several of them in this guide, but they all have the same benefit in mind: making standardised content available in an easily retrievable way and then assisting legal experts to use that content in a way that makes optimal use of their time.



CHAPTER 01

What's the problem?

In this chapter, we investigate the traditional drafting process so we can identify its failings and see how clause libraries can help. If you are an experienced legal expert who has already experienced this process many times over, you are welcome to skip to chapter 2 (before the PTSD kicks in).

Story time: Lucille drafts a license agreement

Meet Lucille.



Lucille is a junior associate in a prestigious law firm, with a little under two years of experience under her belt. She is no complete rookie, but she does not have a wealth of knowledge to draw upon when engaging in legal drafting.

Lucille works for Rob, a partner at the law firm, who hands her a new assignment: “Create the first draft of a license agreement, taking into account that:

- our client is the licensor (so the draft should be licensor-friendly),
- the subject matter is a trademark license, and
- the license should be non-sublicensable, remunerated and globally applicable”.

In a real-life scenario, the instructions would likely be much more exhaustive. For the sake of brevity, let’s focus on these three.

Rob thinks back on his own experience drafting these kinds of documents, takes a quick dive in his email inbox, and provides Lucille with a precedent document to start from. From there on out, it’s up to her.

Let’s see how she fares.

We will be talking quite a bit about “legal nuance” in this guide. We hope the term is self-explanatory. If that is not the case: legal nuances are all those (subtle) elements in a clause or document where different options present themselves to the drafter and where the latter has to make a choice in light of the legal (or commercial) position they find themselves in.

Examples include:

- Employment law – employee type
 - Full-time or part-time
 - White-collar or blue-collar employee
 - Independent or employee status
- Corporate law – SPA pricing methodology
 - Fixed price
 - Locked box
 - Completion accounts
- Data protection law – party relationship
 - Controller – controller
 - Controller – processor
 - Processor – sub-processor

Nuance 1: “Our client is the licensor”

Lucille immediately realises the precedent that Rob gave her is an outright disaster. It’s an old contract dating back 16 years ago, before a major legislative change shook up the legal landscape, so many of the provisions are outdated. Furthermore, the document was drafted for a previous client which did not act as the licensor but instead as the licensee.

There is often a curious disconnect between senior legal experts and junior legal experts when it comes to precedent documents. The former – thanks to their years of experience – are all too eager to believe that “they have something like that lying around.” The latter cannot find that material if it does not exist, and lack the experience to find it if it does.

It’s evident that Lucille will need to find some replacement clauses to deal with the invalid clauses and the fact that she needs to make this document more licensor-friendly.

She starts by going through her own email inbox to see if she has ever worked with a document like the one she is currently drafting, but she has limited experience and so limited information to draw on. She takes a quick look through the organisation’s document management system — but since she cannot draw upon her own experience, she has trouble finding the right keywords to find the right document. She spends a lot of time analysing several of these documents to see if the required legal nuance is there. Many coffee cups later, she finally manages to cobble together a few clauses that are useful, but is forced to resort to a complete rewrite for the rest.

Lucille takes a quick look at the amount of time she has already spent and the nerves creep in. “Surely it’s not supposed to take this much time?” As a young and ambitious lawyer, she is keen to prove herself to the partner in charge, so she decides to write off an hour she has spent on the work already.

Nuance 2: “The subject matter is a trademark license”

Intellectual property licensing tends to be similar enough in that a trademark license can have a lot of clauses in common with other kinds of intellectual property like copyright or designs. Nevertheless, some idiosyncrasies remain, which Lucille now has to deal with.

Nervous about the time she has already wasted trying to find the right clauses, Lucille finds the courage to go knock on some doors. She explains her situation to a few senior colleagues and asks whether they can assist by sending over a few clauses or a precedent contract with the trademark-focus she is after.

She’s in luck: one of her colleagues, having experienced this process many times before, has begun to keep a rudimentary database of precedents and clauses. The colleague in question sends her over some material and Lucille thanks her lucky stars that she doesn’t have to underbill again.

Nuance 3: “The license should be non-sublicensable and remunerated”

Lucille’s trademark licensing agreement is coming along nicely. She has already completed two of the three assignments given to her. She now simply has to tidy everything up to change the scope of the license as per Rob’s instructions.

Fortunately for her, the document contains the following scope clause:

2. SCOPE

2.1. The Licensor grants an exclusive, **sublicensable**, worldwide, **remunerated**, license to the Licensee.

She makes the necessary changes to the words “sublicensable” and “non-remunerated” and she’s set.

Being the detail-oriented lawyer she is, Lucille subjects the document to a final “sanity check”. She has been through this process of sanity-checking before and made a list of elements to watch out for:

- spellcheck and grammar check
- consistent numbering
- consistent styling
- correct use of terminology (she doesn’t want to refer to the client for which the precedent was created!)
- cross-references all still functional (spoiler alert: they weren’t, and she was forced to find and dive back into the precedent document she got the offending clause from to find out what clause it was supposed to reference, losing another 30 minutes)

With her sanity check out of the way, Lucille proudly presents the document to Rob...



...who finds several fatal flaws within 5 minutes of reviewing.

- Lucille had altered the “Scope” clause, but neglected to make any additional changes throughout the document.
- There was still a clause setting out the rules by which sublicensing could occur, which should not have been allowed in the first place.
- There was no clause on payment modalities, which is of course crucial if the license is remunerated.
- There was still a clause discussing “the Territories” where the license applies, despite the license being globally applicable.
- Lucille essentially forgot that the legal nuance she introduced in the scope clause did not carry over to other parts of the document.



CHAPTER 02

How can clause libraries help?

With Lucille's drafting challenges in mind, we can now look at some of the ways a clause library could have helped to avoid some issues and streamline the process.

Speed

Providing a single source of truth where users can find relevant clauses on relevant topics helps to speed up the time spent searching the right clause. This can be done either on the basis of keywords, an orderly folder structure, or different kinds of “metadata” assigned to individual clauses.

That said, searching is only one part of the drafting equation. Some specialised tools also offer ways to deal with tweaking, styling, and optimising by allowing you to add different kinds of flexibility to your clauses.

Education

There is a common misconception in the legal world that providing clause libraries to junior legal experts like Lucille stifles their learning process. Many assume that the trial-and-error process Lucille has to go through each time she drafts a contract is the only way to truly learn how to become a legal drafter.

Not only is this idea highly unfair to clients (they are essentially paying for Lucille’s ‘tuition’ in the school of legal drafting), it is also patently untrue.

Empowering your colleagues with clause libraries is an ideal way to give them a bird’s eye view of the different routes they can take and to get them to think critically about legal nuance as a result.

Suppose Lucille had been using a clause library that offered her the option to explicitly search for the kind of license she was instructed to look for, complete with all the necessary legal nuance. Not only would she not have made the mistakes she did, but she would have been in a better position to appreciate the different ways in which that legal nuance presents itself, and would be better positioned to spot gaps in her document (and even the library itself) going forward.

✓	Benefits	4	Mitigating risk
1	Speed	5	Consistency
2	Education	6	Language versions
3	Compliance	7	Access control

Compliance

Diving into precedents to find the right clause carries serious risk. Lucille was fortunate enough to have caught the fact that the precedent document provided to her was outdated, but not everyone might be as perceptive as her. In the future, this issue could surface again.

For example: had Lucille been presented with the outdated clauses in a clause library, then she could have flagged them in the library and warned her colleagues who would use this material in the future. In the current setup, the precedent will not be altered and when a new lawyer comes along, he or she will still be presented with this material. Who is to say whether they will be as perceptive as Lucille?

Mitigating the risks of precedent-based drafting

Another danger in using old content is that provisions written especially for a client's unique situation may creep into the new one. Central management of clauses in a dedicated library allows you to extricate clauses from their source document and make them available in a more versatile manner.

The danger of precedent-based drafting

It's a well-known fact that lawyers are taught to look to the past: "what has court X said about this topic? What has professor Y written?" This way of thinking is not without its dangers though.

Research on that topic performed on a database of over 12,000 M&A agreements reveals that traditional, precedent-based drafting leads to "a high level of [...] unnecessary and ad hoc edits that appear to be cosmetic rather than substantive. [...] This leads to] haphazard and inconsistent lawyering as lawyers add significant amounts of extraneous information to each deal and inadvertently retain deal-specific information from prior deals."

Consistency

Consistent content – It's always preferable to have a document contain a clear thumbprint of the organisation rather than the individual lawyer drafting it. Law firms especially will want to ensure some consistency in the content and look-and-feel of their documents. Creating a single repository of content for all team members is an important step in the right direction.

Advanced legal drafting software will allow you to create intelligent clauses that immediately streamline such things as terminology and grammar. For example: say you insert a clause from your clause library into a contract that identifies the parties as "Supplier" and "Customer". However, the clause itself assumes the parties are identified as "Service Provider" and "Client", re-

spectively. The terminology that should be adopted by that clause is the terminology used by the rest of the contract in which it is inserted. Rearranging this terminology and grammar takes additional lawyer time and review.

Consistent layout – Dedicated clause library tools allow you to avoid storing the layout of the clause together with the content of the clause. This is important to avoid having to perform a lot of clean-up work when inserting clauses, or to allow clauses to automatically adapt to the styling of the document as they are inserted. But even in “do it yourself” clause libraries like the spreadsheet-powered library (see below), separation of style and content is possible.

Language versions

In jurisdictions that work with multiple official languages, finding the right clause takes on a whole other dimension. Suppose you are based in Spain and have found the perfect clause only... it's in English, and the document you want to draft should be in Spanish. Sure, you can translate it on the spot. But you will likely be reinventing the wheel again. Furthermore, your colleagues will not benefit from the translation exercise you have performed unless they get lucky and stumble upon your document in the future as they are searching for a precedent.

Dedicated clause library tools situated at the more powerful end of the spectrum will not only allow you to store multiple language versions of a clause, they will also provide machine translation functionalities to facilitate a quick translation process.

Access control

On the one hand, your drafting knowledge should be accessible to those who need it. On the other hand, you want to present only relevant content to users (e.g.: employment lawyers rarely have need of a corporate lawyer's clauses).

Clause library tools should allow you to set up these silos. Not only do they improve the experience for end-users, they mitigate the risk of unwanted changes being made to the clauses themselves.



CHAPTER 03

What tools can I use to build clause libraries?

Clause libraries recently (re-)entered the spotlight as the latest shiny feature to add to any piece of legal drafting software. But you don't need to purchase expensive drafting software to create a useful library.

In this chapter, we take a look at the different types of clause libraries that are out there, including the ones you can build right now with technology you have lying around.

What tools can I use?

The document-powered library

01



02

The shared drive-powered library

The spreadsheet-powered library

03



04

Dedicated clause library tools




05

Document automation tools



The “do it yourself” single document approach`



Option 1: The document-powered library

What?

The first, and most basic, approach to store your drafting knowledge is gathering everything in one big document. This has the advantage that all your knowledge is centralised in the same place and is easily searchable. The downside: this document will probably become chaotic, slow and unwieldy once the file becomes too large.

How?

- Create one file to store your library in.
- Use headings to adopt a structure that makes sense to you. The most popular options are:
 - Per legal domain (e.g. commercial law, employment law, corporate law). Useful if you work in multiple domains.
 - Per document type (e.g.: shareholders agreement, power of attorney, corporate resolutions,...). Useful if you work in a specialised field of law.
- Add new clauses as you find them, making sure to group them according to the chosen structure for easy retrievability.

Optionally add comments to individual users to guide them in choosing the right clauses.

Make your document available to team members by storing it in a shared location, such as a central folder in Office 365, a Sharepoint intranet site, Dropbox, and so on.

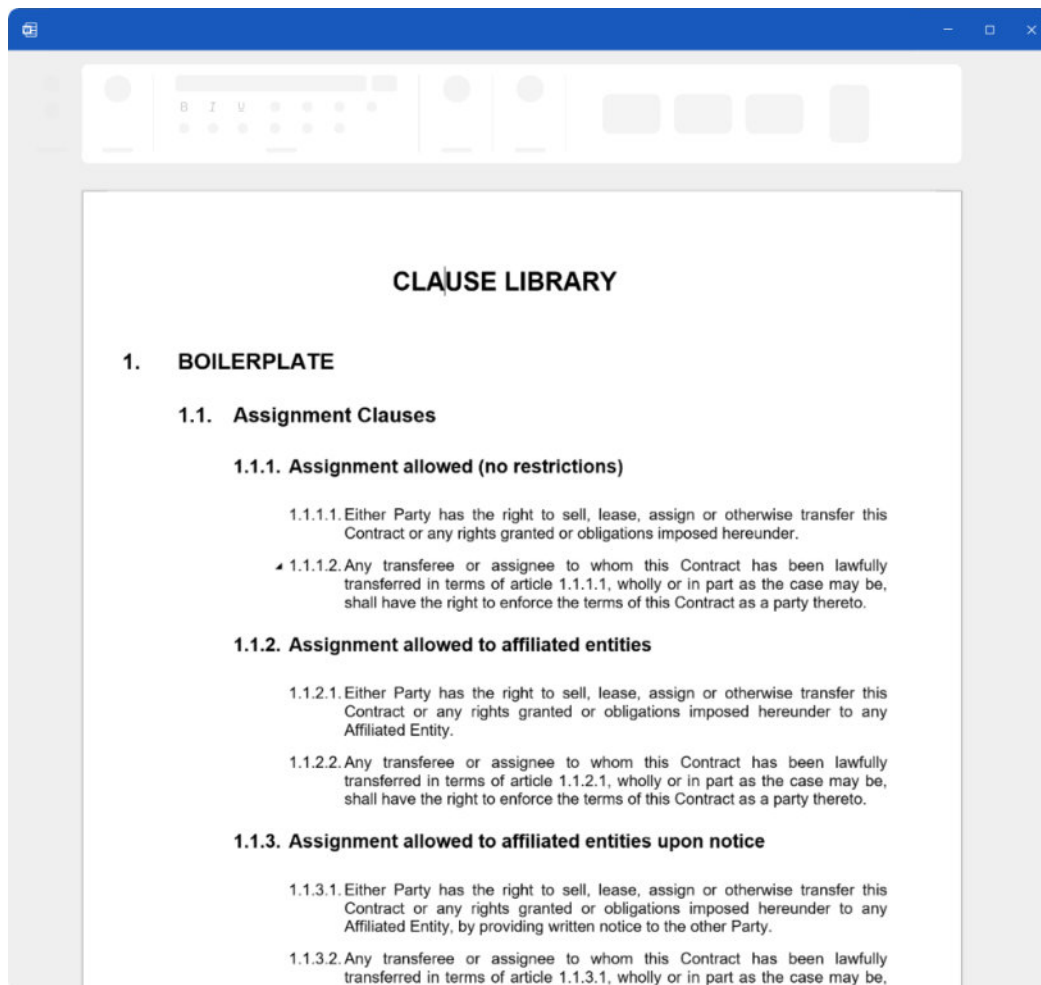
The File of Truth

This story may sound familiar to you: many organisations rely on that one senior lawyer who keeps a history of clauses in a single document file.

That person becomes the go-to person for everyone in the firm looking for a clause. The lawyer in question somehow finds their way around "their" file - often hundreds of pages, organically grown and in complete disarray.

Unfortunately, this solution is not scalable, as it is almost impossible for others to find anything in this file. Another unfortunate drawback of this approach is that when this person leaves the firm, the knowledge leaves as well.

PROS	CONS
<p>Cheap – This approach likely requires zero financial investment since it utilises a tool your organisation should already have access to.</p>	<p>Messy – As your library grows, it will likely be difficult to keep the structure intact. Also, every document ultimately, inevitably succumbs to styling corruption.</p>
<p>Easy – You don't have to learn how to use a new tool.</p>	<p>Limited – While this library may help you find clauses easier, it does nothing to ensure consistency of style, terminology, grammar, language, etc. when you actually use a clause in practice.</p>
<p>Quick – You can get started right now. Just open up an empty document and start building!</p>	<p>IP protection – If the intellectual property of the organisation is contained within a single document, it runs the risk of having departing colleagues easily take that knowledge with them.</p>





Option 2: the shared drive-powered clause library

What?

Another approach to building your clause library is to keep your drafting knowledge in a set of separate small documents and structuring them using hierarchical folders. Here, we trade in the above “quick-and-easy” approach for something a little more structured and scalable.

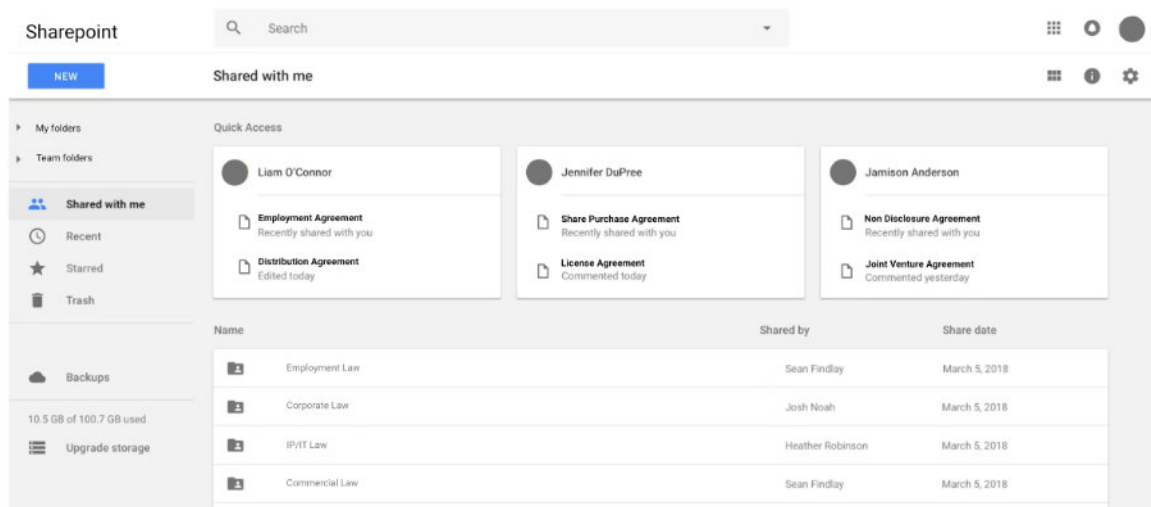
How?

- Create a dedicated space on a drive you share with your colleagues.
- Again, adopt a structure that makes sense to you. This time, however, use individual folders as the base of that structure instead of headings in a document.
- Upload individual document files to the appropriate folders which can either contain single clauses or different variants of a single clause.

PROS	CONS
<p>Collaboration – This approach is much more suited to sharing knowledge with colleagues due to the standardised structure. Furthermore, this approach allows organisations to set up the necessary access rights (e.g.: as an Employment lawyer, you don’t need access/edit rights to the Corporate corner of the library).</p>	<p>Limited – this approach already comes a long way in helping everyone in the organisation (not just the curator of the library) to find the relevant content, but it still falls short on assisting (more junior) lawyers – suppose a lawyer encounters a folder containing 40 non-compete clauses. How does the organisation ensure they use the right one? Ideally, metadata are assigned to the clause to show the different legal nuances at a glance (see below).</p>
<p>Structure – The shared drive-powered library is ideal for setting up a standardised structure which also makes it easier for other users to come in and deposit knowledge into the library.</p>	<p>Setup – It takes more time to set up a shared drive-powered library than a document-powered library.</p>

Location search – An additional dimension of searching for relevant content is created – not only keywords but also location inside the library. This makes it even easier to find the relevant content.

Slow – This approach can be a bit more slow to navigate. The folders in the drive are ideally filled with individual document files (which take a longer time to create and open), and if your folder structure becomes large, you will find yourself doing a lot of clicking to get to where you need to be.



Option 3: the spreadsheet-powered clause library

What?

A third low-tech approach is to use a spreadsheet to store your clauses. This approach combines easy searchability with additional functionalities to create structure and guidance for end users. Clauses can be stored in different rows throughout different sheets, and can have all manner of additional information attached to them in related columns.

How?

Create a single spreadsheet file, optionally subdivided into different worksheets for different domains or different types of documents.

- Create a set of columns where you store the desired information on individual clauses. Popular examples include:
 - Name of the clause to succinctly identify it
 - Title of the clause
 - Content of the clause
 - Legal (sub-)domain
 - Type of document
 - Description of the clause
 - Legal comment on how/when (not) to use

Legal attributes (e.g.: length on a scale from 1–5, aggressive or balanced, identification of a favoured party, company standard or fallback,...)

PROS	CONS
<p>Augmented intelligence – The multi-column approach allows you to transform clauses into more than static bits of text, by augmenting them with additional information. Highly useful if you want to create contract playbooks or focus on helping users make the right choices when drafting.</p>	<p>Size – Collecting all of your material in one spreadsheet will inevitably at some point overburden the application of your choice as your library grows. This can lead to crashes and lags.</p>
<p>Structured & scalable – This approach combines both the easy searchability of the document-powered library with the structure and scalability advantages of the drive-powered library.</p>	<p>Unnatural use of spreadsheets – Spreadsheet tools are not designed to work with large blocks of text. For example: individual words or placeholders cannot be highlighted, and items such as cross-referencing, numbering, styling, etc. are not supported.</p> <p>Furthermore, Excel displays a maximum of 1024 characters (i.e., roughly half a contract page). This means that large clauses must be split over different cells.</p>

Excel functionalities – A wealth of spreadsheet functionalities not native to word processors or drives present themselves here.

For example: columns allow you to sort for specific attributes, while Excel-formulas allow you to search for clauses that have specific metadata assigned to them.

IP protection – If the intellectual property of the organisation is contained within a single document, it runs the risk of having departing colleagues easily take that knowledge with them.

Clause identifier	Clause body text	Folder	Description	Length	Favoured Party
Introduction to the Parties	The present Contract is entered into between [name], with address at [address], incorporated under the laws of [country] and with identification number [company number] hereinafter referred to as the Manufacturer and The present Contract is entered into between [name], with address at [address], incorporated under the laws of [country] and with identification number [company number] hereinafter referred to as the Distributor with either party being referred to as "a Party" or, cumulatively, "the Parties".	Introduction	Introduction of two parties to a distribution agreement from the assumption that there is one distributor and one manufacturer and that both are legal entities.	3	None
Purpose of the agreement	The Manufacturer hereby authorizes the Distributor to distribute the Product in accordance with the terms of this Contract. The price for the Product shall be as established in the price list as attached to this Contract.	Scope	The Manufacturer assigns the Distributor to perform the necessary services. Price set out in an annex to the main agreement.	1	Manufacturer
Price	Unless where stated explicitly otherwise, any prices stated in this Contract shall be: - stated in [currency]; and - exclusive of any taxes and duties (such as value added tax). The price for the Product shall be [amount].	Price	Price set out as a fixed amount in the contract itself. Useful for smaller-scale agreements.	2	None
Price (fallback)	Unless stated explicitly otherwise, any prices stated in this Contract shall be exclusive of any taxes and duties (such as value added tax).	Price	Price set out as a fixed amount in the contract itself. Useful for smaller-scale agreements.	1	None
Payment modalities	Payments due must be made in the applicable currency through bank transfer on the bank account number mentioned on the invoice of the Manufacturer and including the required references. The Distributor shall provide proof of payment to the Manufacturer when requested to do so. Payment through set-off is never possible, unless otherwise agreed in writing. All payment obligations are non-cancellable and all amounts paid are non-refundable except where explicitly agreed otherwise in writing. In the event of a dispute with respect to a part of an invoice, the part of the invoice that is not disputed shall be paid in accordance	Payment	Payment through bank transfer with set-off not being allowed. Heavily in favour of the Manufacturer.	4	Manufacturer

Dedicated clause library tools

Examples include: *ClauseBuddy*, *ContentCompanion*, *Woodpecker*

Over the past couple of years, several dedicated tools have begun popping up that are geared towards searching and using your clauses in ways where the above mentioned tools fall short.

These tools allow you to create a clause library – typically from within MS Word – which you can populate and use with just a few clicks. Different tools come with different flavours, of course, so below is a list of required features you may consider if you are looking for a dedicated clause library tool.

Keyword searches

Any dedicated clause library tool should allow you to search on the basis of keywords.

Note, however, that solely searching on the basis of keywords may not always yield a useful result.

For example: many clauses that do not directly deal with the topic of liability will use this word in some way. Search for a liability clause using the keyword “liability” may yield a lot of noise.

Folder structure

A folder structure provides an additional dimension in which users can search for the right clause.

While keyword searches are primarily useful if you already know exactly what you are looking for, a carefully considered folder structure is much if you are looking for inspiration on clauses to add.

More advanced libraries allow for hierarchical (“nested”) folders, and some even allow to cross-reference between different folders. In light of the complexity of the legal topics, the importance of such features should not be underestimated.

Populating the library

Two options can be discerned: manual population vs artificial intelligence (AI) powered population.

While manual population is more time-consuming upfront for the library author, this approach allows you to build a more qualitative library, augmented with all manner of legal metadata.

AI-powered population allows you to immediately draw all clauses from a range of documents. While this approach consumes practically no time upfront for the library author, it focuses on quantity over quality and may run into serious compliance issues if personal data contained in the clauses is not scrubbed (see more in Chapter 05). In addition, AI-powered population typically yields less accurate information, and little to no legal metadata or ordering, users of the library will typically spend much more time when using the library.

Augmentation

Most clause library tools – DIY or dedicated – already capture some information automatically in relation to your library or clauses. For example: if you open up the “File” tab in MS Word/Excel, you will immediately find information such as the date of creation, original author, date of last modification, user who made the last modification, etc.

The added benefit of dedicated clause libraries is that they can give you additional metadata to sort and search content.

Styling

Some clause library tools augment clauses by default in such a way that they can automatically adapt themselves to the styling of a document when they are dropped into it.

Upgrade path

Some clause library tools also offer an upgrade path from basic storage of clauses to full-on document assembly or automation later on (see below).

Analytics

Tools that track the usage of particular clauses offer an easy way of figuring out the most useful clauses in the library. However, this does not paint a full picture, because they cannot track which clause makes it into the final document after discussion with the client or negotiation with the counterparty.

Clause subscriptions

Some dedicated clause library tools allow you to make (parts of) your clause library available externally – this is particularly useful for legal service providers looking to innovate service delivery to their clients.

Document management and document automation/assembly tools

Examples include: ClauseBase, ContractExpress, Docassemble, HotDocs,...

While many clause library tools are standalone, some also function as part of a larger tool.

Document management tools allow for archiving, tracking, and following up on existing documents. These tools primarily benefit from clause libraries in that they can draw analytics on how many times a clause was used or renegotiated (allowing you to create so-called “heatmaps” of clauses).

Document automation tools focus on generating entire documents from scratch based on a flexible template. Clause library-augmented document automation tools allow you to sync clauses over multiple documents. A change made to a single clause can then ripple through all the documents it is used in, greatly easing template maintenance. They are also sometimes called document assembly tools.

Powerful document automation tools allow you to create entire clause generators, by capturing all the legal nuance a given type of clause can have in a single file. These clauses act as real chameleons – able to take on any colour to suit the environment they are placed in, from both a style and a content perspective.

To see such a chameleon clause in action, take a look at this [dispute resolution clause generator](#).



CHAPTER 04

How do I start building a clause library?

In the previous chapter you have learned what clause libraries are, and how legal experts can benefit from building one. If you're still with us – terrific! That means you are ready to start taking your first steps into a new way of legal drafting.

But knowing that you need a clause library is not the end game. The next step is to start building one. What are the questions you should ask yourself before you start? What do you put in your clause library? And how do you structure it?

Question 1: Who needs access?

Specialisation in legal teams has been the norm for the past few decades. Few legal experts nowadays can claim to do it all (how many legal experts do you know that juggle labour disputes and securitisation transactions)?

This means that the average legal team will have quite a diverse collection of clauses. In order to make the relevant content available to the right legal experts, it will be necessary to create silos. Think about how your organisation is structured – which departments, which legal matters? Those will likely be a good starting point to set up your base silo structure, but some cross-departmental groups may also be necessary (e.g.: providing legal services to the aviation industry often sits at the intersection of corporate law, employment law and financial law).

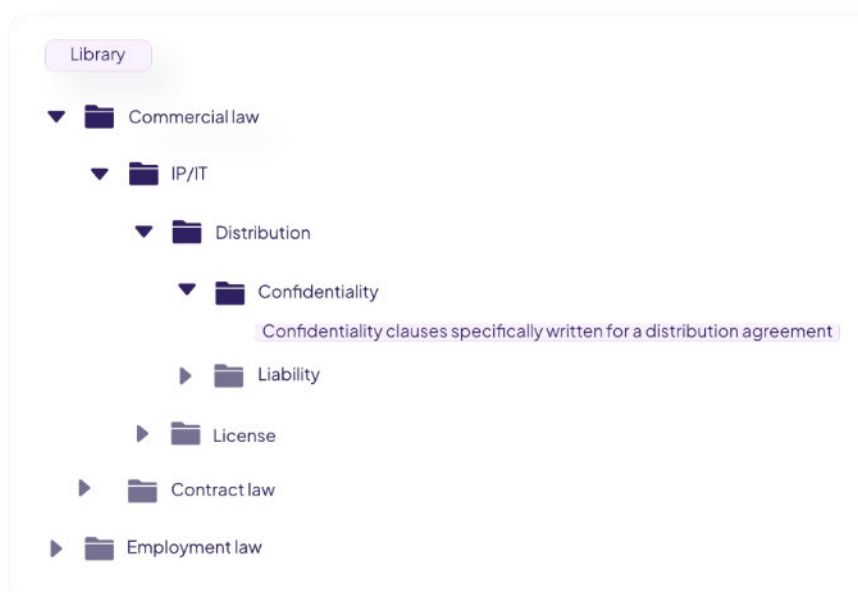
Geographical location can be another factor to play a role in creating these silos (or even sub-silos). The IT law team of an international law firm may wish to provide certain clauses to both its Brussels and Paris offices, but cordon off some more jurisdiction-related clauses in sub-silos.

Deciding which and how many silos you need is the first step of setting up your library. From there, you can further set up your folder structure.

Question 2: How to set up my folder structure?

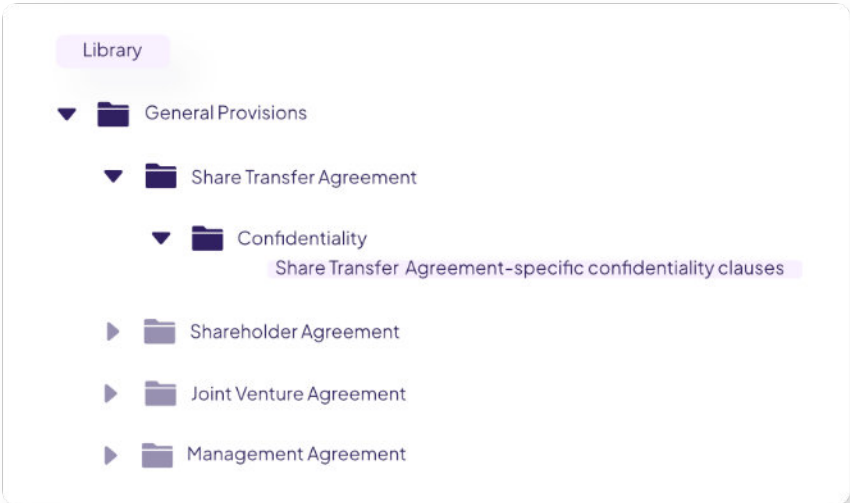
Option 1 - Based on legal domain

The first way to structure your clause library is by starting from the legal domain to which the clause relates. From there you can further divide your folders depending on the type of agreement and on the subject matter of the clause. In the example below, we have illustrated what this structure can look like.



Option 2 - Based on the document type

Another starting point can be the type of document in which a clause would typically be used. In this case, you will have a general subject matter at the first level and then further divide your clauses depending on the document you would likely find them in. This approach is especially useful if you are focusing on a highly specialised area of law.



As we have mentioned above, the way in which you structure your library is your choice. There is no perfect approach – both options have their advantages and disadvantages. If there is a rule to structuring a clause library it is this: “be consistent”.

Whichever route you choose to take, know that you will likely need to refer to other parts of the library from time to time.

Say you practice corporate law and chose to implement option 2 above:

- Corporate
 - General Provisions
 - Share Transfer Agreement
 - Shareholder Agreement
 - Joint Venture Agreement

For all of three documents, you might consider creating a separate folder for Confidentiality clauses. You could create an agreement-specific folder “Confidentiality” for each individual agreement, but you will likely be able to reuse a lot of material between the three. Instead, consider creating references (also sometimes called “shortcuts” or “proxies”) from one location to another. That might look something like this:

- Corporate
 - General Provisions
 - Confidentiality
 - Share Transfer Agreement
 - Confidentiality (reference to General Provisions)
 - STA-specific confidentiality clauses
 - Shareholder Agreement
 - Confidentiality (reference to General Provisions)
 - SHA-specific confidentiality clauses
 - Joint Venture Agreement
 - Confidentiality (reference to General Provisions)
 - JVA-specific confidentiality clauses

Question 3: Which clauses do I add to my clause library?

Many organisations have written thousands of clauses in the past. How do you create a clause library out of them without spending hundreds of hours on it? It’s a common misconception by legal teams that their clause libraries should contain all of the clauses the organisation has ever created in the past, contained in documents widely dispersed over email inboxes, document management systems, shared drives, etc.

There are several issues with this assumption. We’ve collected the most important ones below.

Ease of use

Suppose your organisation has drafted 500 share purchase agreements in the past. Would you want the clauses from all 500 documents in your library? Here are just a few issues to take into account:

- What to do with overlapping clauses?
- What to do with outdated clauses?
- What to do with redlined clauses?

Or take the example of a confidentiality clause. A quick search through your firm's documents will likely yield hundreds, if not thousands of different results. A quick search online will likely yield thousands, if not tens of thousands of results.

Now think about the different ways you can actually write a confidentiality clause. Most likely, it will take a position on one of the following legal nuances:

- Unilateral vs mutual
- Aggressive vs balanced
- Small scope or large scope
- Designation of authorised recipients vs no additional recipients
- Explicit penalty or no explicit penalty
- ...

In total, there will probably be around 5 to 10 different legal nuances per clause type. Those legal nuances then tend to be combined into typical clusters, e.g. a short unilateral aggressive clause with explicit penalties versus a balanced mutual clause with no penalties. The amount of clusters (combinations) will differ somewhat, but is typically quite manageable — e.g., about 50 in practice, of which you will probably only use of a subset.

Conversely, when you would create an inventory of all the confidentiality clauses in all the contracts your organisation has made in the past, you will probably end up with hundreds if not thousands of examples. The underlying reason is that there are thousands of cosmetically different ways to write a clause, even though legally speaking many of those would boil down to the same clause (i.e., the same combination of legal nuances).

Forcing yourself to search through all those cosmetically different clauses every time you want to insert a confidentiality clause will lead to frustration and manual rewrites.

The trick is to find those versions and only upload those legally different clauses. This requires an element of human creativity and expertise that AI is currently not capable of replicating.

Compliance

Law is constantly evolving. New jurisprudence, new legislation, new interpretations, etc. What is common practice today might be ridiculed tomorrow.

Like Lucille, you will likely not be interested in using clauses that were written before the last 10 years. Why, then, would you want to capture those clauses?

You already have access to all these clauses through other tools

If your organisation has been around for at least a few years, you will likely have access to a case management system, filing system or a document management system that contains a treasure trove of information. Any such tool worth its salt will already have search functionalities that allow you to search for text on the basis of keywords. In other words, an easy way of finding text already exists.

Of course, clauses found through such generic systems are not structured in any way, with many disadvantages as a result:

- As a lawyer who worked on a specific file, you will know where to look for clauses you drafted previously, but what about your colleagues?
- You will have to manually extract clauses and “clean” them, both in terms of layout and in terms of terminology.
- Even more problematic: no context is provided in selecting the right clauses. Do not underestimate this problem, as discussed below.

Do not underestimate the “missing context” problem. It’s not so difficult for legal experts to assess what is explicitly written in a clause — most legal experts can probably quickly decide whether an explicitly written element is appropriate or not.

Much more difficult is to think about what is not written in a clause. For every clause you extract from an old file, you must be aware that there could have been reasons for omitting legal elements.

Take, for example, a penalty in a confidentiality clause. When a penalty is present in the clause and you don’t think it is appropriate for the contract you are drafting, you can decide to either remove the penalty, or take another clause without a penalty.

Compare this to the situation where you extract a random clause from a random old contract, and the penalty happens to be missing there, e.g. because one of the parties deliberately removed it as part of the negotiation. Are you sure that you will think about re-inserting it when necessary? Are you confident that your junior colleagues will do so?

Most legal experts will probably admit that they will think about the 3 or 4 most primary elements of a clause when they are in clause-hunting mode. But whether they will also think about the secondary elements, is much less certain.

Question 4: How should I augment my clauses?

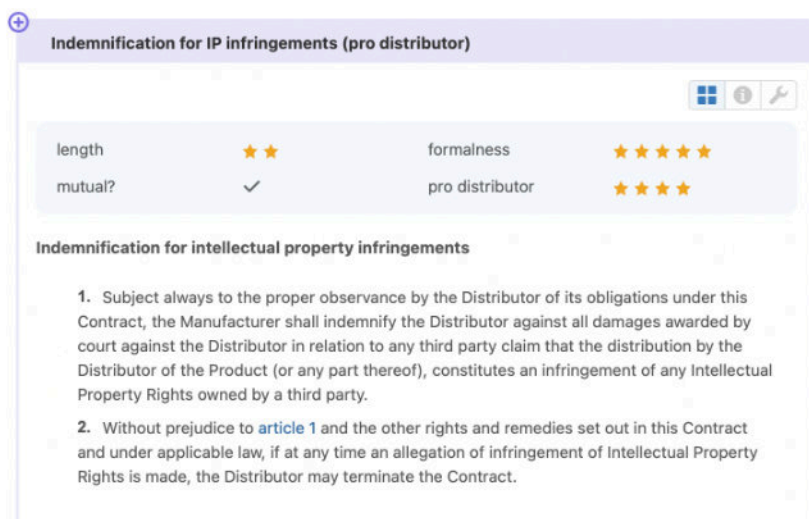
When building a clause library, you are probably interested in doing more with your clauses than just storing bits of static text. Any element added to a clause effectively augments the clause, i.e. increases its usefulness.

There are many different types of augmentation.

A first augmentation is to provide some background about the origin of the clause, e.g. the specific client or transaction from which it was extracted. Particularly when your clause library allows you to search on this information, this can be very useful to include. In practice, however, the origin is mostly useful for personal libraries, as opposed to libraries that are used in a team. After all, knowing that a clause was created for “Project Alpha” or “Client Smith” may be useful information for the person who was involved in those situations, but may be useless (or even confusing) for the junior lawyer using the library afterwards.

More useful is to include information on how or when to (not) use the clause, i.e. to provide some guidance to the library user. Depending on your library software, you can either draft such information as free text, or use a standardised format.

- An example of free text would be “This is a fairly one-sided clause that is appropriate to use if we are acting for a party with significant negotiation power. Be aware that this clause cannot be used towards consumers in France.”, or — in an in-house setting — “Never use this clause in contracts for product line X, as it is missing feature Y, which would not comply with internal drafting policy Z.”



- An example of a standardised format is to assign certain “attributes” or “legal nuances” to a clause, e.g. “mutual” as opposed to “one-sided”, or “retail sector” as opposed to “construction sector”, or “4/5 on the length scale”. While some products provide a predefined list of attributes, others allow you to completely customise those attributes.

Third, you can also include legal references, such as case law, legal doctrine or references to statutory provisions associated with the clause in question. You may even want to include hyperlinks towards external websites or internal intranet sites.

Question 5: Who do you need to build a clause library?

It doesn't take a village to build clause libraries. You can perfectly create a clause library on your own, for your own. But even if you work in a team, you will find that there are really only two types of hats that can be worn.

The curator	The user
<p>Like a curator in a museum, the Curator has two main responsibilities:</p> <ul style="list-style-type: none"> • Decide which material makes it into the library. • Update and maintain the existing material. 	<p>The User's primary role – as the name might imply – is to use the clauses contained in the library to draft legal documents. In that capacity, they also act as useful sources of information for the Curator, as they can flag potentially interesting additions to the library, which the Curator can then decide should be included or not.</p>
<p>Anyone can be designated as the organisation's Curator (multiple Curators may even be operating at any given time). The best Curators are typically:</p> <ul style="list-style-type: none"> • Designated knowledge managers. • More senior lawyers who understand the different kinds of legal nuance in their drafting field of choice. • Experienced paralegals. 	<p>Identifying who acts as a User in your organisation is a little more straightforward – it's anyone who is not a Curator.</p>

BONUS question: How time-consuming is building a clause library?

We've talked about the important of quality vs quantity and how it's much better to have 10 standardised, legally nuanced clauses of a given kind than a deluge of 10.000 unstructured clauses.

Extrapolating on that, it's important to realise that building a clause library can be a gradual, organic effort and doesn't need to be Herculean feat of uploading hundreds of thousands of clauses in one sitting. Getting your blueprint in place is crucial. After that, you can just add clauses to the library as you come across them and gradually watch it grow.



CHAPTER 05

The role of AI in building clause libraries

Legal professionals are busy people. They often cite time commitment as their main impediment to building a clause library – not just the time to add the clauses but also to maintain and update them. The question typically goes “Can artificial intelligence (AI) not build this for me?”

For the sake of simplicity, we label all software tools that automatically build clause libraries as “AI-based”, because that’s the terminology used by most marketing departments. There are actually quite some differences between “AI”, “data mining”, “supervised learning” and “unsupervised learning”, but we are trying to keep it 101.

Indeed, a number of AI-powered tools (Draftwise, Genie AI’s SuperDrafter, Henschman) have popped up recently that allow you to extract vast amounts of clauses from the documents your organisation has lying around. Then there’s also the tried and true LawInsider, which provides access to hundreds of thousands of publicly available clauses – more variety than your own organisation could ever hope to produce.

So, is this technology up to snuff?

Keyword-focus

AI-based clause library software tools do a good job in letting you search for a clause using keywords. Such keyword searches are mature and widely used in every industry. Accordingly, you can be confident that when you are searching for “employee liability”, you will indeed get a list of clauses that contain those words. Some of the more advanced tools will even automatically “cluster” clauses on the basis of the words in the clauses, so that similar clauses can be automatically associated with each other.

What’s really attractive about these tools, is of course that they require no time investment from legal experts. Your IT department will install the tool, and you are immediately good to go.

If all you need is to occasionally find that one clause you had written in the past, this seems like the perfect solution.

So what’s the catch?

Finding unique keywords. Keyword-based searching works great with unique keywords (“bankruptcy”, “force majeure”) or specific expressions (“material adverse effect”). Unfortunately, these are a minority, because many clauses tend to be made up of a relatively small number of unique keywords. For example: words like “party”, “confidential”, “material”, “liable”, “obligation” are repeated in many different types of clauses.

Missing labels

The most interesting keywords will often be missing. For example, when a corporate lawyer wants to insert a “Texas shootout” clause, a useful keyword might be “shootout”. Unfortunately, that word is most likely not literally present in the clause text itself, so an AI-tool will be at a loss to find a clause like that using a keyword search. Similarly, a commercial lawyer will frequently want to find a clause with a low liability cap or a maximal responsibility carveout, but real-life clauses don’t literally mention such qualifications for obvious reasons. As a result, you will have a hard time finding such a clause, because it will typically be made up of a series of popular keywords (“share”, “buy”, “sell”, “price”, “liable”, and so on).

Insufficient volume

You need to feed the software gigabytes of data to make it somewhat smart. As a rule of thumb, you need to have at least several thousand different versions of each relevant clause in your jurisdiction to allow the software to completely independently figure out that two pieces of text are actually variations of a same clause (near-identical copy/pastes don't count). Even the largest UK/US law firms would perhaps only come close to this amount when including material received from clients and counterparties, or when including intermediate versions during negotiations — but it's probably not wise to randomly include such material.

When the volume of data is not sufficient, you will not get Google-like search satisfaction. Instead, you will have to dig through long lists of chaotic, half-baked results.

Mental processing time

Law is a profession of words, and drafters are all experts in massaging text. As a result, it can take a minute (particularly with long clauses) to interpret a clause and determine what it is really saying and, and what is deliberately hidden. If all you get back from your software consists of bare clauses, you will spend a lot of mental processing time understanding each search result. Over time, across a team, all this time spent interpreting search results quickly adds up — undermining the efficiency you were pursuing.

NDA issues

When clause libraries are automatically constructed, quite some confidential information will inadvertently get embedded into the database. As a result, everyone who uses the library will see confidential bits & pieces flying around when scrolling through the search results. This can easily breach typical confidentiality obligations in NDAs (“access must be restricted on a need-to-know basis”) or local bar rules.

Data breaches

When you mass-feed old files to your software tool — particularly when hosted by a third party — you are reusing documents containing often highly sensitive personal data for a purpose (your drafting comfort) that is incompatible with the initial purpose (handling a client file). With EU authorities [increasingly upholding strict interpretations](#) of the GDPR, such “repurposing” of personal data could be considered a [fundamental breach](#) of the Regulation. While automatic “scrubbing” will remove some personal data, it will never reach the extremely high anonymisation level that [data protection authorities require](#).

In conclusion, are AI-based clause libraries interesting?

To give the lawyerly answer: “it depends”.

If you are a solo lawyer and mainly want to search through your old clauses, they can be a good fit. You know your own material well, so you know which keywords to use. And you can probably quickly reconstruct the context that is missing in each search result. Confidentiality issues should be minimal, and if you work outside the EU you likely won't be impacted by the GDPR.

If you are part of a legal team, the assessment becomes different, because you will have to balance the advantage of near-zero upfront legal expert time with the downside of spending relatively more time scrolling through — and mentally interpreting — the search results. You will also have to consider your risk position with regards to potential compliance issues.

In the end, the assessment probably boils down to how you look at a clause library.

- If you are mostly interested in optimising the occasional search for that one clause that you just know to have written in the past, and can still remember parts of it, then an AI-based library tool will be a great help.
- If you are more interested in overall workflow enhancement, sharing knowledge and optimising the quality of your team’s output, then an expert-curated library is more for you. You will spend some legal expert time upfront, but will typically get multiple times that investment back.

	AI Library	Expert Library
Upfront effort IT-team	Several hours or days	Negligible
Upfront effort legal experts	Negligible	Distributed over a period of time
Search speed unique clauses	High	High
Search speed common clauses	Low	High
Interpretation time	High	Low
Context & guidance	Missing	Augmented clauses
Compliance risks	High	Low

BONUS ROUND: How does AI work?

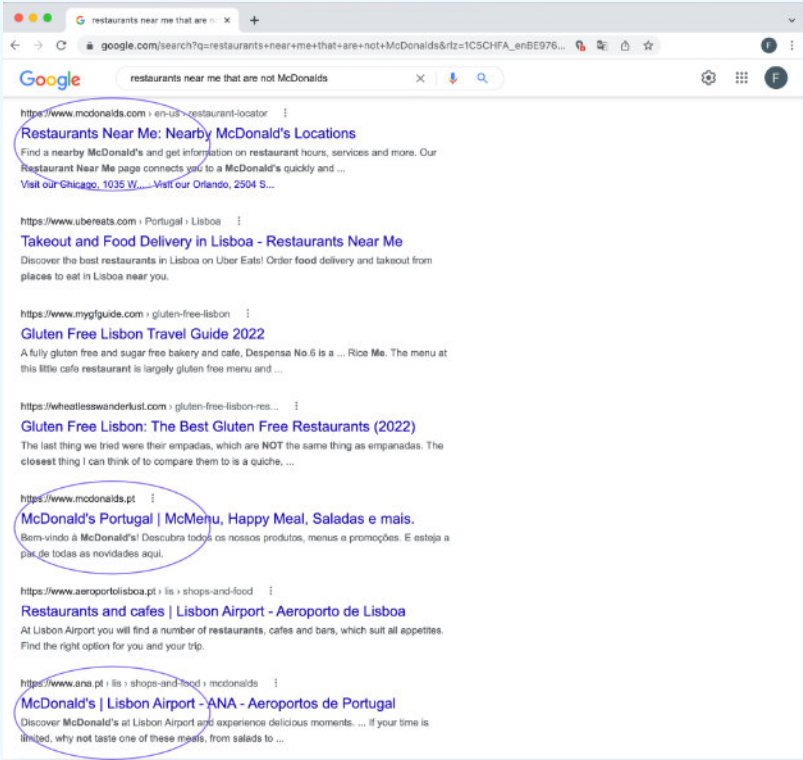
If you want to really understand the technical side of how AI plays a role in these situations, see the breakdown below. It’s not necessary to help you build clause libraries, but we’d hate to deny anyone the opportunity to geek out a little.

AI has made enormous strides over the past few years, from [GPT-3 writing articles](#) to [Tesla’s self-driving cars](#). Even more established technology like machine translation or online search engines like Google have seen their functionality vastly improve in recent years.

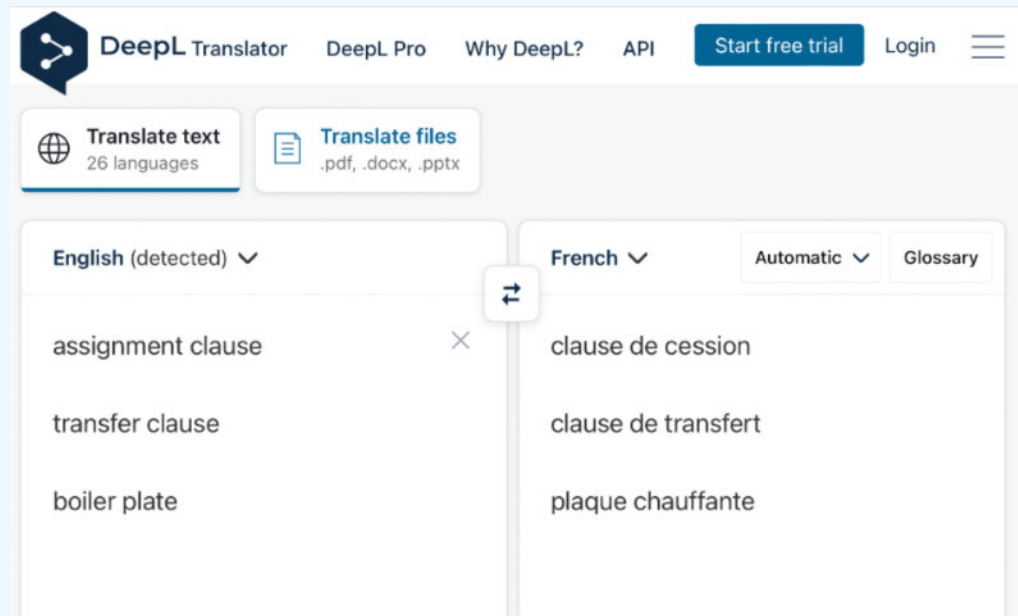
The number one factor of success for all of these AI-powered technologies has been the sheer volume of data that has been made available to help train the AI. GPT-3 had to parse over 45 terabytes worth of text — that’s about 30 billion pages — with a capacity of about 175 billion machine learning parameters to get to where it is now. Tesla had to subject its AI to billions of miles of driving before it could reach a level of technical viability on highways. Similarly, machine translation engines incorporate millions of language pairs from which they distil their knowledge (e.g., all the statutory provisions translated by EU parliament translators).

If huge amounts of data are available, such "narrow" applications of AI are like magic, as anyone will have to admit when using modern translation engines such as [DeepL](#), when cruising on the highway with Tesla, or when realising that Google autocompletes your search query as if it can read your mind. However, none of these technologies actually understand what is going on: they merely calculate statistical correlations, but with sufficient data available such correlations feel like magic.

If you want a good example of how easily you can trick the AI of the most advanced search engine on the planet, try googling “restaurants near me that are not McDonalds”. You will be surprised to see that almost all restaurants listed will be... McDonalds.



Similarly, even advanced translation engines have difficulties understanding context. In the example below, any legal expert will understand what “boiler plate” means. The translation engine translates it to “plaque chauffante”, i.e. a heating plate...



Besides the required volume, there is a second hurdle to overcome: comprehension. Even software that claims to be “self-learning” is not doing any actual “learning”, in the sense of how human beings would acquire new skills. The software will merely look for patterns that have statistical significance. And even with large amounts of data, this can go completely wrong.

Take the story of the AI that was [trained to distinguish between wolves and huskies](#).

The software learned to identify them successfully, achieving very good accuracy with the sample images fed to it. But when practiced on other photographs, the software completely failed. The reason? From the software's perspective, the difference between a wolf and husky was the presence or omission of snow in the background. When confronted with a picture of a wolf in a snowy landscape, the software immediately assumed that it was looking at a husky...

Applied to the legal sphere: let's say you are trying to teach AI to recognise a governing law clause. You can feed it millions of examples, but the exact lessons it will pull from that exercise are anyone's guess. If a few too many of these governing law clauses refer to "Delaware" as the applicable law, it might assume that governing law clauses are all clauses that contain the word "Delaware".

When it comes across a party introduction clause including a legal entity incorporated in Delaware, it may flag it as a governing law clause. When it comes across a clause selecting "New York" as its governing law, it may not recognise this as being a governing law clause.

A final problem is that it can only learn from what is present in the examples given to it. This is particularly problematic for a clause library, because what is not written in a clause is probably as important — and sometimes even more important — than what is explicitly written down.

This can become a particular hurdle for AI in continental Europe, where statutory provisions (e.g., articles of the Civil Code) tend to function as "fallback" provisions that will apply even when clauses are silent about that topic. The typical example is good faith in contract law: in many European jurisdictions, parties do not need to mention it for parties to be obliged to act in good faith towards each other. From the AI's perspective, there will be a significant difference between clauses that explicitly mention good faith, and those that don't. From a continental lawyer's perspective, it will depend on the context, the drafting style, the length, and so on whether a clause that explicitly mentions good faith is indeed different from the same clause that omits any reference to good faith.



You are all set!

Armed with the information above, you are now fully equipped to harness clause libraries to empower a new form of drafting – one focused on value-add, both for your colleagues and for your clients.

Quick overview

If you are still digesting the information above, consider using this quick overview to figure out your next steps:

- Identify the basic structure of your library. Typical options include:
 - Order by legal domain
 - Order by subject matter
- Identify the kind of information you want to augment your clauses with (so-called “legal metadata”). Some common options are:
 - Length
 - Clause type (e.g.: “company standard”, “alternative”, “fallback”,...)
 - Favoured party (formulated either as “mutual?” or “Seller/Purchaser-friendly”)
- Based on the above, identify the tool you wish to use to set up your clause library. Some basic DIY options are likely already available in your organisation through the Microsoft O365 or Google Workspace suites, but dedicated tools like ClauseBuddy and ClauseBase can really take your drafting process to the next level.
- Identify the users in your organisation who will need access and the roles they will fulfil. As a rule of thumb, you will probably want 1 Curator for every 10 Users

If you want to see a fully constructed sample library in practice, then sign up for a free [ClauseBuddy account](#) and gain access to a collection of curated boilerplate clauses and clause generators on a variety of topics!



About ClauseBuddy

ClauseBuddy strives to raise a new generation of lawyers. Providing both basic and advanced solutions to answer every legal professional's needs. It is our mission to let everyone discover the possibilities of software-assisted drafting.

Explore more on www.clausebuddy.com