Creating an artificial intelligence for NDA evaluation

A.P. Engelfriet, M.Sc., LL.M., European patent attorney

Legal ICT / JuriBlox BV / Vrije Universiteit

a.engelfriet@juriblox.nl / +31 20 663 1941

Electronic copy available at: https://ssrn.com/abstract=3039353

Abstract

The non-disclosure agreement (NDA) or confidentiality agreement is a staple of the business community, in particular the high-tech and IT business. However, there is something of a gap in the market between what legal professionals need to review an NDA and what businesspersons are prepared to offer for such services. A machine learning system is constructed that can recommend whether a given NDA is acceptable or not from the perspective of the user. The system divides NDAs into sentences regarding different topics and uses a first support vector machine to assign topics. Sentences are then grouped by topic and qualified (e.g. as strict or relaxed, or as standard, broad and limited) by a set of second support vector machines. The system was trained on 80% of 304 source documents, and performance was enhanced using various techniques, resulting in a machine learning system that can make correct recommendations on NDAs with 87% accuracy (F1=0.86).

Keywords: artificial intelligence, Bayesian classifier, confidentiality agreement, non-disclosure agreement, support vector machine

Electronic copy available at: https://ssrn.com/abstract=3039353

Creating an artificial intelligence for NDA evaluation

1. Introduction

The non-disclosure agreement (NDA) or confidentiality agreement is a staple of the business community, in particular the high-tech and IT business. Originally created for protecting trade secrets, NDA's are now routinely used to cover confidential exchange of any business information, from prototype designs to customer lists or proposals for new business ventures (La Fontaine & de Ly, 2006). It has been suggested that the NDA is the most-used legal document in business.

Due to their ubiquitous nature and routine use, NDA's are generally regarded as a standard document. From a legal perspective, one couldn't be more wrong. Drafting or reviewing a non-disclosure agreement is an art (Wiggers 2011, Anderson & Warner, 2016) and therefore requires significant amounts of work to get right (Miller, 2006). This translates into a significant price tag to such a review, which is hard to grasp for an average businessperson who just wants to hear if (s)he can sign that NDA or not. As a result, there is something of a gap in the market between what legal professionals need to review an NDA and what businesspersons are prepared to offer.

In the past decade, the use of machine learning has received increased attention in the legal field, e.g. for semantics extraction (Biagioli, Francesconi, Passerini, Montemagni & Soria, 2005), document classification (Quaresma & Gonçalves, 2010) or automated summarization (Yousfi-Monod, Farzindar, & Lapalme, 2010). More generally, see Lodder & Mommers (2007). Limited research however has been published on the application of machine learning specifically to contracts (Curtotti & Mccreath, 2010). More work has been done, e.g. de Maat & Winkels (2008) or Biagioli et al (2005), on the automatic classification and analysis of laws.

A likely reason to the dearth of research in this specific field is the very broad nature of the legal contract. Literally any topic can be addressed in a contract. This makes it difficult in general to design machine learning systems for contract analysis beyond the very general level. Chalkidis and Michos (2017) for example are able to extract various general contract elements, such as contracting parties, contract period and governing law, from arbitrary contracts using a combination of machine learning and post-processing rules. More research is summarized in Curtotti & Mccreath (2010).

As it turns out, NDAs are in essence very limited in their scope: certain information is to be exchanged for a given business purpose, and this information is to be kept confidential for a certain period. This limited scope makes them well-suited to automated analysis. There are after all only so many ways to say that information must be kept confidential and that leaks shall be judged by a court in California. This makes likely the hypothesis that a machine learning system that can accurately evaluate an NDA can be constructed.

2. Analyzing legal documents

A preliminary question is how to approach the analysis of a legal document. In the practice of law, contractual obligations are structured in clauses, which each may be comprised of one or more sentences (Biagioli et al., 2005). Typically, a clause deals with one subject only. Clauses can be grouped by subject, e.g. a clause setting a contract term and several clauses providing options for termination of the contract can be grouped under the subject of term & termination.

To provide a legal analysis, the clauses of the NDA in question must be evaluated one by one. However, it is hard to identify clauses as such from an unstructured text document. It is on the other hand easily possible to extract individual sentences from a textual document. This led to a two-stage design, following the suggestion in Kim (2017) that there may exist

underlying segmentation of sentences in a document, and perhaps this partitioning might be intuitively appealing (e.g., each group corresponds to a particular sentiment or gist of arguments). In the first stage, sentences are classified into a general category (e.g. definition, security requirement or termination), and in a second stage, sentences in the same category from a single document are grouped and assigned a qualification within the category (e.g. strict definition, moderate security requirement or termination at any time). In this context, categorization means broadly assigning a topic, while qualification means assigning a specific variety within that topic. For example, when all sentences categorized as "defining when information is confidential" are combined, it becomes possible to qualify this definition clause as e.g. very limited or very broad.

Two-stage classifiers are of course known in complex multi-category classification tasks (Giusti & Sperduti, 2002). Such classifiers deal with the problem that category boundaries may be hard to draw, especially when classification within a single category is desired. The first stage makes a broad classification, which is used as an input in the second stage to make a more precise qualification. What is novel in the present approach is that it is used for textual analysis, first at the sentence level and then at the clause level.

The system generally operates as follows. In the creation stage (next section) a collection of source documents is split into sentences, after which each sentence is classified broadly in one of plural general categories such as "Defining what is confidential information", "Deals with export control" or "Sets a term on confidentiality obligations" (see Table 1 for a full list of categories and explanations). This input is used to create a first-stage, general classification model. Next, sentences from the same document that are in the same category are collated and assigned a qualification within the category (e.g. strict, moderate or neutral) to create second-stage, category-specific models. Following similar reasoning as Yan,

Dobbs & Honavar (2004), it is believed that qualifying clauses within a known category produces better effects than trying to identify and qualify a clause from scratch.

In actual usage, an input document is received and converted to plain text, and each sentence is classified using the first-stage model, after which sentences from the same document that are in the same category again are grouped and qualified using the second-stage model. The meaning of the qualification, and thus the legal opinion on the clause, follows from a profile separately provided by the user. For example, a person intending to receive confidential information would benefit from having a strict definition of confidentiality, as it limits his liability for accidentally sharing certain information. A person intending to share confidential information however would rather have a very broad definition. Various response tables are needed to map a qualification to a response given the profile. In the example, if the profile reveals the person intends to receive information, the response for a strict definition of confidentiality would be acceptable, while the profile for a person intending to give information would elicit a rejection of such a strict definition.

3. Data set creation

Essential to creation of a machine learning model is the creation of a data set. The data set for the present project was created following the steps of Kotsiantis, Zaharakis, & Pintelas (2007). The main steps involved are (i) source collection, (ii) data preparation, (iii) feature selection, (iv) algorithm selection and (v) training and testing the classifier. Steps (i) through (iv) are discussed in this section.

Source collection

In this experiment, source collection mainly involved acquiring a large number of non-disclosure agreements. Next to the author's personal archive (with, where necessary, customer consent properly obtained) the main source was public internet services offering model documents. In a timespan of about five weeks (March/April 2017) 375 source documents were acquired.

During initial review a clear distinction between business contracts and employee contracts quickly emerged. Employee non-disclosure agreements were often very one-sided, used very different and broad language and generally were one page at the most. Given the goal of the project, such agreements would not be very useful. Therefore 55 employee agreements were eliminated from the source set. A further 16 documents were eliminated because due to copy protection mechanisms their textual content could not be copied with lawful effort, leaving a final 304 documents for review.

Data preparation

Data preparation in this project mainly involved assigning category labels to the sentences of the source documents, and subsequently qualifying sentences grouped by category. A first command-line tool was created that parsed each document into a collection of sentences, presenting each sentence for manual categorization (see Figure 2). Headings were detected using a number of simple heuristics (e.g. "Sentence starts with 'Article' followed by one or more digits and a period, or "Sentence is at most six words and four or more of those are capitalized"), and the current heading was recorded together with the category of each sentence. The first sentence of each document was skipped as it generally contained no more than a title.

Which categories to include was a difficult task in the data preparation stage. An initial 16 documents was reviewed with the aim of creating classes on the fly, each sentence

being assigned a category based on the professional experience of the author. Subsequently the categories were reviewed and pruned, resulting in 42 initial categories. During further review, the number of categories grew as new types of sentences appeared. In the end, the number of categories was limited to a nice round 64.

After completion of the document-by-document categorization of sentences, a second command-line tool was created to allow review of all sentences per category (see Figure 2). This proved effective in identifying miscategorization, and more importantly in organizing categories. For example, initially separate categories for IP ownership and for licensing rights sentences were created. However, during review it was found that these two categories had significant overlap, leading to the decision to unify the two into a single category.

Further, during review a scrubbing of company-specific and/or personal data was performed. For example, some documents contained a specific person's name as authorized recipient of confidential information or contained a company address in the party identification section.

A third command-line tool was created to group sentences by category and assign qualifications within the category (see Figure 3). This tool is discussed in more detail below in Section 5.

Feature selection

Feature selection is the process of identifying valuable features (e.g. words or word combinations, but also metadata such as sentence number or length) and removing as many irrelevant and redundant features as possible. For textual documents, this generally involves well-known processes such as stemming, removal of stop words and case-insensitive processing.

The nature of the NDA makes it hard to extract relevant features other than the sentence or clause itself. Often, an NDA is nothing more than a sequence of clauses that each

deal with separate subjects. However, many NDA's in the source collection turned out to have headings per clause. These were extracted using simple heuristics as described above, and recorded together with the sentences that appeared under them.

Algorithm selection

Today a wide variety of machine learning classifiers is available, both for in-house use and as cloud-based service solutions. Often-used classifiers include naïve Bayesian classifiers and support vector machines. For the specific task of text classification, support vector machines are especially well-suited (Joachims, 1998). A choice was made to work with the machine learning services of BigML. Inc. from Corvallis, OR (USA). According to its website, "BigML is a consumable, programmable, and scalable Machine Learning platform that makes it easy to solve and automate Classification, Regression, Cluster Analysis, Anomaly Detection, Association Discovery, and Topic Modeling tasks." The modeling tools of BigML allow creation of multi-label textual classifications, and models created using the service can be deployed off-line.

4. Sentence classification model creation and improvement

The sentence dataset obtained in the previous section was split in a standard 80/20 training/evaluation division on a per-category basis. That is, for each category 80% of the sentences was included in the training dataset and 20% in the evaluation dataset.

BigML offers both individual decision tree models and ensembles. Ensembles are learning algorithms that construct a set of models and then classify an input using a weighted vote from each model in the set (Dietterich, 2000). As a variation, BigML offers the boosted trees construct (Schapire, 2003), where each model in a number of iterations attempts to correct for errors in the output of the previous model. For the classification stage of the project, ensembles were used as these generally perform better in multi-classification tasks.

Model creation

Three models were created in order to evaluate their respective performances: April, a bagging ensemble or decision forest; May, a random decision forest and June, employing boosted trees. These were the three ensemble options offered by BigML. Default settings were used in each case.

Upon creation, each model was subjected to a standard evaluation using the 20% evaluation dataset. The results are included in Table 1. While various quality measures are available, the F1 measure is recommended for sentence classification (Khoo, Marom & Albrecht, 2006). Using this measure, it is clear that June, the boosted tree model, performed significantly better (F1 = 0.64) than both April and May (F1= 0.44 and 0.45 respectively) in all aspects. April and May thus were discarded, and further evaluation and improvement focused solely on June.

Model improvement

The first step towards improvement is to study the confusion matrix (de Maat, E., Krabben, K., & Winkels, 2010). This reveals which categories bear most false positives or suffer the most false negatives. The misclassified results in question were reviewed individually. This revealed three areas of improvement: reclassification, adding synthetic data and adding contextual flags.

Review and reclassification

Firstly, it was found that some 30% of the evaluation data was misclassified by the author (or at least, that the model found an equally valid classification for the item). These classifications were corrected. While there always is a risk of overfitting when a dataset is adjusted based on test output, care was taken to ensure that any reclassifications were objectively justified given the original criteria of the class. During further evaluations, similarly classification errors were identified and corrected as needed.

Second, several classes were conjoined into one as their mutual criteria turned out to be too vague for a clear distinction. The original class labels were retained in the dataset so as to not lose this information should it be necessary for later use. Class labels were replaced in post-processing.

Third, compound clauses were split. Compound clauses are defined as clauses that enumerate three or more separate issues in a single sentence,[1] which resulted in more-or-less random classification between these issues. The source processing tool was rewritten to output such enumerations as separate sentences, adding the preamble of the sentence to each separate sentence. Source data review also revealed several sentences with two separate issues, typically separated with the word 'and'. However, splitting sentences on this or similar transitional words proved too complicated, as more than 40% of all sentences contained the word 'and'.

Third, additional features were added by augmenting each sentence in the dataset with the feature of its relative position in the source document, expressed as a percentage. It was expected this feature would contribute to better classification, as in legal documents it is common to put certain clauses at the beginning (e.g. a definition of the parties) and others more towards the end (notably the so-called boilerplate). Additionally, the word count of each sentence was added.

The above modifications resulted in the Webbigail ensemble which provided a significant improvement (F1=0.68) over June (F1=0.64).

---

[1] For example: "Recipient shall (a) treat all Confidential Information with the highest care; (b) only permit persons having a clear need to know access; (c) evaluate the Confidential Information at its own risk; (d) comply with relevant export regulations; (e) indemnify and hold harmless Discloser from any damages in connection with usage of the Confidential Information; and (f) waive its right to a jury trial." This sentence would be split into six sentences, each starting with "Recipient shall".

Adding synthetic data

As a general rule, a machine learning model becomes more accurate if more data is supplied. In this experiment, only xxx documents were available and given the nature of the source documents it is hard to obtain a large number of additional documents with reasonable efforts.

A promising route was the approach suggested by Chawla, Bowyer, Hall & Kegelmeyer (2002) to create synthetic examples by combining features from multiple existing items to form new ones. The term 'synthetic' here refers to the fact that the items are based on real-world data, but in artificial variations. For example, in the context of legal clauses, one might take sentences containing the word 'permission' or 'harm' and create clauses using the synonym 'consent' or 'injury', and vice versa. Manual creation of additional documents and/or sentences was undertaken but proved very labor-intensive, even with the addition of dictionary inputs. A creative solution was found in deploying a Markov text generator[2] using the existing sentences as input to synthesize sentences from classes with low performance. This provided an additional input of 1,517 sentences, and the resulting ensemble Cynthia improved markedly in F-measure (F1=0.73) compared to Webbigail (F1=0.68).

Adding contextual flags

The ensembles used in this experiment all operate on the well-known bag-of-words (BOW) concept, where the words from each sentence are treated individually. While this is remarkably effective, all contextual information from the sentence was lost. A common suggestion is to then create bigrams or longer n-grams as features which contain extra

[2] In the experiment the generator at https://ermarian.net/services/converters/markov/words was used, generating 5.000 words at a chain length of 6 with manual post-processing to eliminate obviously incorrect sentences.

information: "the disclosing party" becomes one single element rather than {"disclosing", "party", "the"} in the BOW approach. However, in general, this approach is not very effective (Bekkerman & Allan, 2004).

It was speculated that for legal documents, this may be different. Legal sentences contain many stock phrases (e.g. "null and void" or "having a legitimate need to know") that effectively can be regarded as single words. Further, certain constructs reveal more information than only the words can express. For example, a common construct is to define terms by providing a definition and adding the term in parentheses and quotation marks: *The parties wish to confidentially discuss a potential merger (hereinafter: "the Purpose").* This contextual information – having a definition in this sentence – is lost in a BOW approach.

Using regular expressions, such constructs were identified and appropriate flags were inserted into the dataset (following Dave, Lawrence, & Pennock, 2003 and Sathyendra, Wilson & Sadeh, 2016). The constructs in question concerned definitions of the parties, confidential information, the public domain, applicable law and competent venue. Standard phrases were replaced with single words (e.g. "Confidential Information" was turned into CONFIDENTIALINFORMATION) and other constructs had a flag appended (e.g. "hereinafter 'Discloser'" was turned into "hereinafter 'Discloser' PARTYFLAG"). In a related improvement, words such as 'no' and 'not' were identified with a separate flag as it turned out the automatic stemming feature of BigML removed these words. The resulting ensemble Wilhelmina provided a significant improvement (F1=0.86) over Cynthia (F1=0.73).

5. Clause qualification model creation

Building on the experiences with the sentence classification model above, clause qualification models were developed. As a first step, a selection of categories had to be made for which further qualification is desirable. Based on the author's personal knowledge, a

selection was made of key categories whose content (rather than presence or absence) is material to the question if an NDA is acceptable or not. For each of these categories (identified in table 2 under column 'Status') a second-stage model was developed. The term 'flavor' is used to qualify the content.

It was found that the second-stage models performed slightly worse than the first-stage model. This was to be expected: the amount of training data is much lower when sentences are combined. To enhance performance, the second-stage models were complemented by a set of regular expressions that catch common phrases suggestive of particular flavors. For example, in the categories for duration and contract term a regular expression capturing phrases of the type "twelve (12) months" or "five years" was quite effective. The second-stage model was used only if the regular expressions were unable to find any results. During testing, this did not result in any mistakes.

As a second step, the categories are divided into three groups. A first category is so important that they – in the wrong flavor – determine that the NDA cannot be signed. In business jargon, these categories are deal breakers.[3]

Other clauses are worthy of notice but not so important that their presence, absence or particular variation would materially affect the evaluation of the NDA. These clauses (identified in table 2 under column 'Key' as serious) are to be flagged, but do not by themselves decide the fate of the NDA. As a rule of thumb, if too many of these clauses appear in the wrong flavor, the NDA should still be rejected.

---

[3] A deal breaker generally refers to a condition or event which causes an agreement to no longer wish to be pursued. It is something that prevents the formation of a contract. It is a term commonly used in the negotiation stage. For example, if two parties are negotiating for a lease and the landlord states that no pets are allowed, it may be a deal breaker for a prospective tenant who is a pet owner. (https://definitions.uslegal.com/d/deal-breaker/)

Lastly, NDAs tend to contain a large amount of so-called 'boilerplate', i.e. clauses addressing standard issues such as the severability of invalid clauses, procedures for sending notices or declarations on how to construct other clauses. These clauses (identified in table 2 under column 'Boilerplate') do not need further qualification and are generally irrelevant in evaluation of the NDA.

6. System evaluation

Having built both classification and qualification models, a simple web-based implementation of the system was deployed. Textual documents could be uploaded through a standard HTML form, after which a server-side process extracted its plain text and categorized each sentence as set out above. The clauses are qualified as explained in the previous paragraph. Figure 4 illustrates an output of the system as a global summary, while Figure 5 shows output at the deal breaker level in more detail.

To create the actual evaluation of the NDA, the qualifications must be put into context. For NDAs this context is the position of the person considering signing the NDA: is this person giving or getting information? As explained in the introduction, a strict clause is generally to the advantage of one party. Thus, knowledge of the position of the party is essential to evaluate the clause.

In the HTML form, this position was requested (as the standard give/get/mutual distinction [Miller, 2006]) and used to retrieve one of three response tables. Each response table provides an explanation and impact code of a particular qualification. An example is given in Table 3 for the category of 'parties', where three flavors (Unilateral, Mutual and Absent) are used. For each flavor an appropriate assessment is given for each of the positions.

The presence of any dealbreaker clauses in an unacceptable qualification according to the response table results in a recommendation not to sign the NDA. In any event, all key

clauses are identified and discussed using the explanations from the response table. For the serious clauses, a negative recommendation is given if 30% of the clauses found has an unacceptable qualification. Boilerplate clauses are hidden in their entirety, and shown upon user request as mere notices.

7. Conclusions

A machine learning system was constructed that can accurately evaluate a non-disclosure agreement (NDA). NDAs are in essence very limited in their scope, which makes them well-suited to automated analysis. The system divides NDAs into sentences regarding different topics and uses a first support vector machine to assign topics. Sentences are then grouped by topic and qualified (e.g. as strict or relaxed, or as standard, broad and limited) by a second support vector machine.

The SVN was constructed as a boosted tree ensemble and trained on 80% of 304 source documents. To compensate for the small source data set, various enhancements were deployed to the data set. Splitting of compound sentences - a particularly common occurrence in legal documents - proved to be rather effective. Surprisingly, adding synthetic data generated by a Markov text generator on existing data gave markedly improved results. Adding contextual flags also significantly improved results. Legal sentences contain many stock phrases ("null and void") and constructs such as in-line definitions which provides additional information that is lost in a standard bag-of-words approach used in typical text analysis. Finally, regular expressions for string matching were used in several cases to assist qualification of clauses. The result is a machine learning system that can make correct recommendations on NDAs with 87% accuracy (F1=0.86).

The system is currently being deployed on the Internet at <http://ndalynn.com>. Further improvements are expected when additional source documents become available in

sufficient numbers. Further study is recommended in particular for the qualification stage, where regular expression matching is expected to provide significant improvements. For example, the ensemble for the venue clause (specifying which court is competent to hear disputes) may benefit from a simple string matching algorithm that looks for city names and thus reveals the actual venue. Other text enrichment options (e.g. Zhang & He, 2015) are also worthy of further study.

References

Anderson, M., & Warner, V. (2016). *Drafting and negotiating commercial contracts*. Bloomsbury Publishing.

Bekkerman, R., & Allan, J. (2004). *Using bigrams in text categorization*. Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst.

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., & Soria, C. (2005). Automatic semantics extraction in law documents. *Proceedings of the 10th international conference on Artificial intelligence and law* (pp. 133-140). ACM.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Chalkidis, I., Androutsopoulos, I., & Michos, A. (2017). Extracting Contract Elements. *Proceedings of International Conference on Artificial Intelligence and Law, London, UK*.

Curtotti, M. & McCreath, E. (2010). Corpus Based Classification of Text in Australian Contracts. In *Proceedings of the Australasian Language Technology Association Workshop 2010*.

Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528). ACM.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, *1857*, 1-15.

Giusti, N., & Sperduti, A. (2002). Theoretical and experimental analysis of a two-stage system for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 893-904.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, 137-142.

Kim, M. (2017). Simultaneous Learning of Sentence Clustering and Class Prediction for Improved Document Classification. *International Journal of Fuzzy Logic and Intelligent Systems*, *17*(1), 35-42.

Khoo, A., Marom, Y., & Albrecht, D. (2006). Experiments with sentence classification. In *Proceedings of the 2006 Australasian language technology workshop* (pp. 18-25).

La Fontaine, M. & De Ly, F. (2006). *Drafting International Contracts*. Brill.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Informatica* 31, 249-268.

Lodder, A. R., & Mommers, L. (2007). Legal Knowledge and Information Systems: JURIX 2007: the Twentieth Annual Conference (Vol. 165). IOS Press.

de Maat, E.; Winkels, R.G.F. (2008). Automatic classification of sentences in dutch laws. In *Legal Knowledge and Information Systems: JURIX 2008: the Twenty-first Annual Conference* (Vol. 21, p. 207). IOS Press.

de Maat, E., Krabben, K., & Winkels, R. (2010). Machine Learning versus Knowledge Based Classification of Legal Texts. In *JURIX* (pp. 87-96).

Miller, J.E. (2006). *Forty Non Disclosure Agreement Review Tips.* Contract Management Magazine. NCMA.

Quaresma, P., & Gonçalves, T. (2010). Using linguistic information and machine learning techniques to identify entities from juridical documents. In *Semantic Processing of Legal Texts* (pp. 44-59). Springer Berlin Heidelberg.

Sathyendra, K. M., Schaub, F., Wilson, S., & Sadeh, N. (2016). Automatic extraction of opt-out choices from privacy policies. In *Proceedings of the AAAI Fall Symposium on Privacy and Language Technologies, Arlington, Virginia, USA*.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification* (pp. 149-171). Springer New York.

Wiggers, W. J. (2011). *Drafting Contracts: Techniques, Best Practise Rules and Recommendations Related to Contract Drafting*. Kluwer.

Yan, C., Dobbs, D., & Honavar, V. (2004). A two-stage classifier for identification of protein–protein interface residues. *Bioinformatics*, *20*(suppl_1), i371-i378.

Yousfi-Monod, M., Farzindar, A., & Lapalme, G. (2010). Supervised machine learning for summarizing legal documents. *Advances in Artificial Intelligence*, 51-62.

Zhang, P., & He, Z. (2015). Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification. *Journal of Information Science*, *41*(4), 531-549.

Tables

Table 1

Relative performance of models

Ensemble	Feature	Accuracy	Avg. precision	Avg. recall	Avg. F1
April	Decision tree	63.18	51.16	42.65	0.44
May	Decision tree	67.86	51.13	44.10	0.45
June	Boosted ensemble	77.16	70.28	62.08	0.64
Webbigail	Cleaned	76.23	72.21	67.54	0.68
Cynthia	Synthetic examples	83.66	77.54	72.54	0.73
Wilhelmina	Added flags	87.21	88.24	82.85	0.86

Note: Accuracy is the number of correctly classified instances in the dataset over the total of instances evaluated. Precision is the number of true positives over the total number of positive predictions. Recall is the number of true positives over the number of positive instances. F1 is the balanced harmonic mean of precision and recall.

Table 2

Categories used in the sentence-level classification

Category	Description	Status	Flavors
parties	Definition of the parties to the NDA	Dealbreaker	Unilateral, mutual
defconf	Definition of confidential information	Dealbreaker	Standard, limited, broad
term	Term of the NDA	Dealbreaker	Short, long, infinite
duration	Duration of the confidentiality obligations	Dealbreaker	Short, long, infinite
public_domain	Circumstances under which information has lost its confidential status	Dealbreaker	Key
need_to_know	Scope of people to whom confidential information may be provided	Dealbreaker	
independent	Addressing the parties' mutual independence	Dealbreaker	
ip	Intellectual property ownership & license provisions	Dealbreaker	Standard, strict

Category	Description	Status	Flavors
security	Security requirements for receiving party	Dealbreaker	Standard, strict
law	Applicable law	Dealbreaker	
venue	Competent court	Dealbreaker	
required_disclosure	Provisions on disclosures required by law or court	Serious	
residuals	Carving out residual information from obligations of confidentiality	Serious	
attorneys_fees	Compensation of attorneys' fees in lawsuit	Serious	
nonconf	Stipulating that certain information is explicitly excluded from confidentiality.	Serious	
press_release	No publicity on the existence of the NDA or the fact that discussions are taking place under NDA.	Serious	
audit	Whether a party has the right to audit the other party for compliance.	Serious	
notification	Procedures for notification of NDA violations	Serious	
copies	Stipulating under which conditions copies may be made from the information.	Serious	
warranty	Any warranties to be given by a party.	Serious	Standard, broad
compliance	Any specific laws with which parties have to comply.	Serious	
liability	Provisions on liability for a party	Serious	Standard, broad
personal_data	Provisions regarding personal data as defined in US or EU data protection legislation.	Serious	
retain_backup	Stipulating that after destruction or return of confidential data, a backup copy may be retained.	Serious	
reverse_engineer	Restrictions on reverse engineering information from prototypes, software et cetera.	Serious	
amendment	Setting out procedure for amending the NDA.	Boilerplate	
assignment	Procedure for assigning NDA to other entity.	Boilerplate	
construction	Rules on construction of the legal provisions.	Boilerplate	
original	Confirming that electronic documents are originals under the law and that multiple copies may constitute originals.	Boilerplate	

Category	Description	Status	Flavors
notice	Where to send required notices under the NDA.	Boilerplate	
representative	Confirming that the entity signing the NDA is an authorized representative.	Boilerplate	
entire_agreement	Confirming that the NDA document is the entire agreement (no side letters or informal deviations).	Boilerplate	
equitable_relief	Confirming that injunctions or other equitable relief may be sought instead of only damages in case of violation.	Boilerplate	
return	Requiring return or destruction upon request and/or termination of the NDA.	Boilerplate	
export	Requiring compliance with laws on export of military, dual-use or otherwise sensitive technology.	Boilerplate	
severability	Confirming that an invalid clause does not affect the legal status of the rest of the NDA.	Boilerplate	
third_party	Confirmig that third parties have no rights.	Boilerplate	
purpose	Recitals, general statements and the like.	Boilerplate	
waiver	Confirming that failure to enforce a clause does not imply a waiver of such clause in the future.	Boilerplate	
agency	Confirming that neither party may act as an agent of the other party.	Boilerplate	

Note: Dealbreakers items are material to the NDA. Serious items should be flagged if present but do not materially affect the NDA (in the legal jargon, are not dealbreakers). Boilerplate items may be ignored. Items that have flavors are qualified further in the second-stage model (section 5).

Table 3

Flavors and their impact for the 'parties' category

Name		Flavor	Use case	Text
parties	Parties definition	unilateral	Give	This NDA defines one party as giver, the other as recipient of confidential information. That works, as long as you are mentioned as discloser of course. ok
			Get	This NDA defines one party as giver, the other as recipient of confidential information. That works in principle, but unilateral NDAs like these tend to be very one-sided. alert
			Mutual	No way! This NDA only defines one party as giver, the other only as recipient of confidential information. That doesn't work with your mutual sharing situation. notok
		mutual	Give	The definition of the parties is mutual: both parties can give and receive confidential information. That's ok if you keep in mind that what they give in return is confidential too. alert
			Get	The definition of the parties is mutual: both parties can give and receive confidential information. While you are the intended recipient, a mutual NDA is good for you as it normally is more balanced than a single-sided NDA. alert
			Mutual	The definition of the parties is mutual: both parties can give and receive confidential information. That's exactly what is going to happen. ok
		absent	Give	There is no definition of the parties AT ALL in this document. Please get some kind of definition in the text, even if it's only the two companies' names. And mark yourself "Disclosing Party". notok
			Get	There is no definition of the parties AT ALL in this document. Please get some kind of

Name	Flavor	Use case	Text
			definition in the text, even if it's only the two companies' names. And mark yourself "Receiving Party". notok
		Mutual	There is no definition of the parties AT ALL in this document. Please get some kind of definition in the text, even if it's only the two companies' names. And add "Each is both Disclosing and Receiving Party". notok

Figures

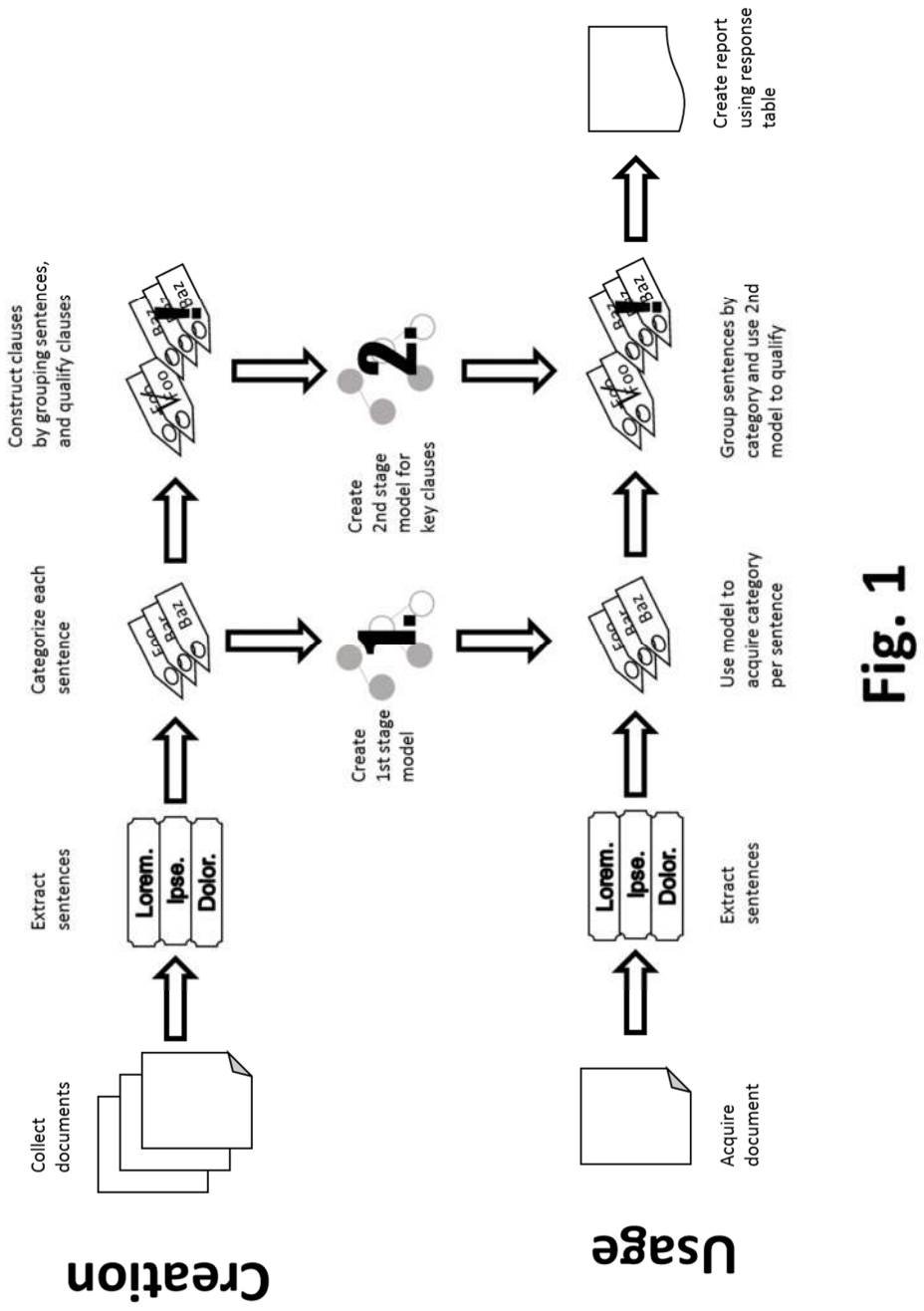

Fig. 1

Figure 1. The general system architecture for both the model creation and usage phases. In the creation phase, sentences are extracted from the collected documents and categorized by topic to create the first model. Sentences on the same topic from a single document are

grouped into clauses and qualified to create the second model. In the usage phase, a single document's sentences are extracted and categorized using the first model. Next, sentences on the same topic are grouped and qualified using the second model. The qualification is used as input for a response table to evaluate the document.

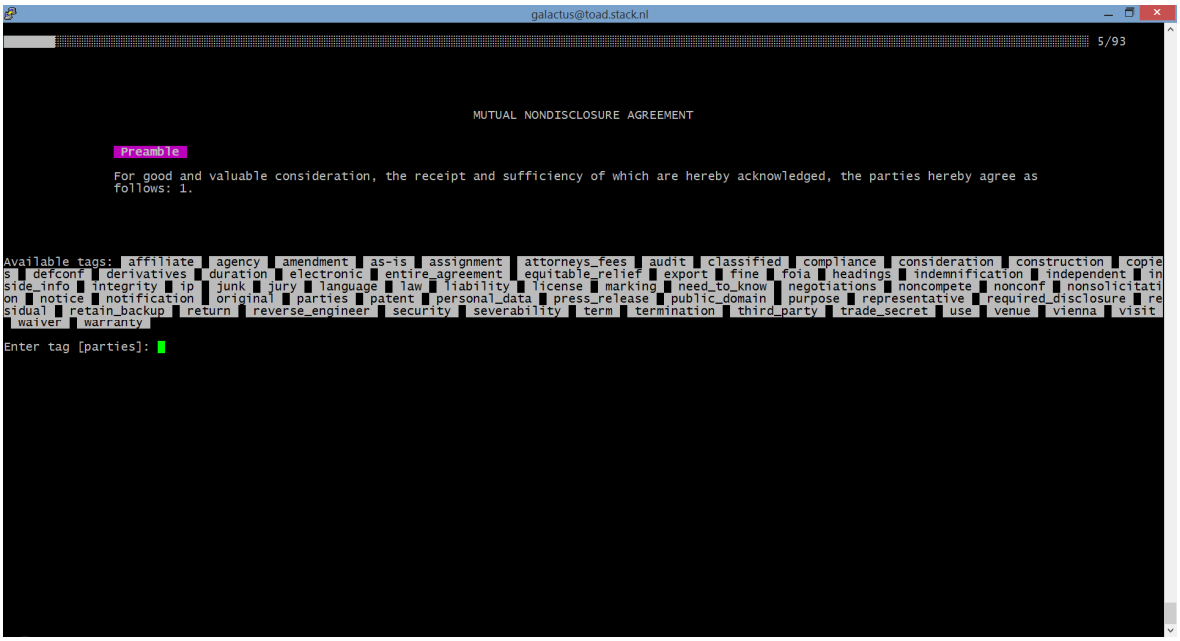

Figure 2. The 'tagger' command-line tool allowed sentence-by-sentence review of an individual NDA.

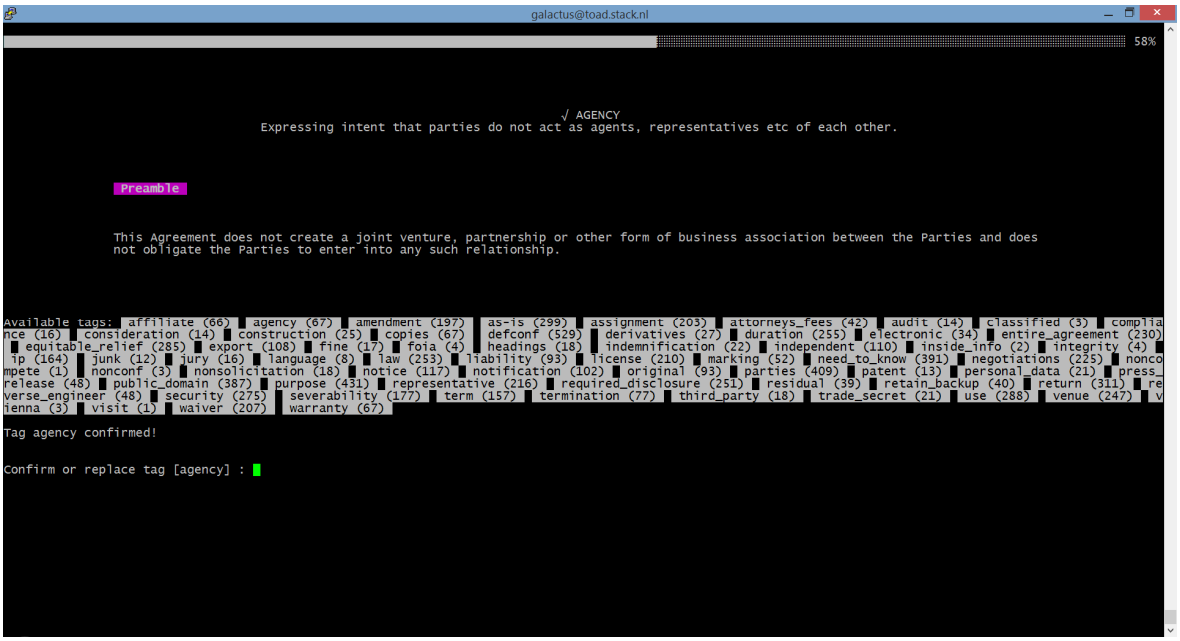

Figure 2. The 'retagger' command-line tool allowed review of all sentences assigned a particular category. In the example shown, the category "Agency" is reviewed.

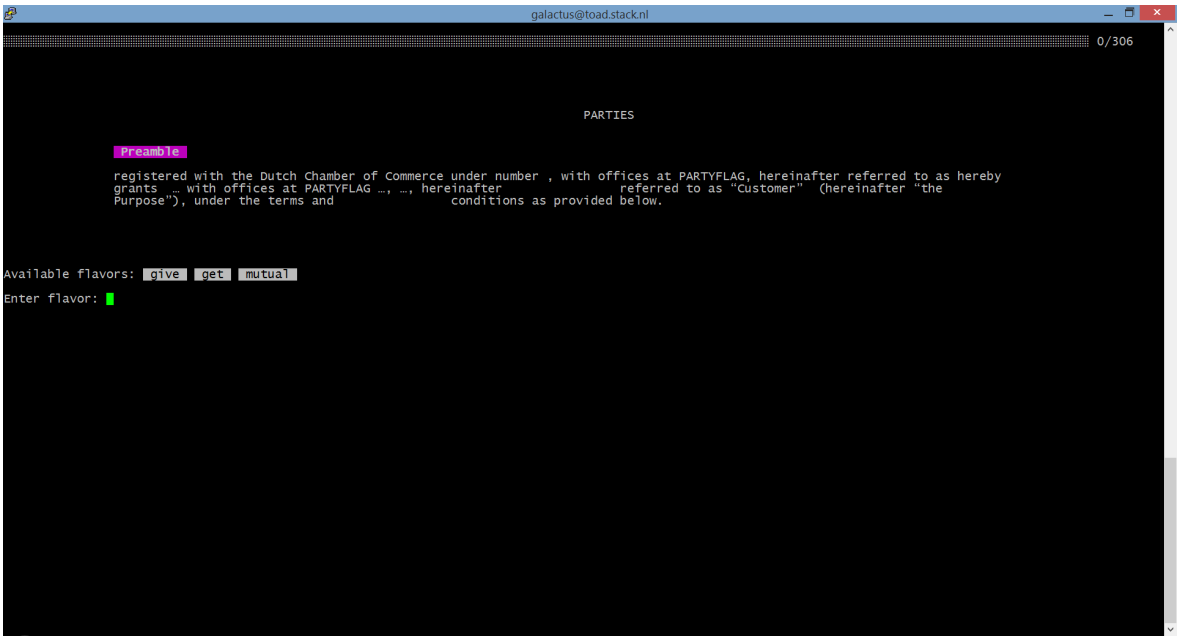

Figure 3. The 'clauser' command-line tool allowed review of clauses comprising sentences grouped by category. In the example shown, a group of sentences in category "Parties" is reviewed for qualification into flavors "give", "get" and "mutual".

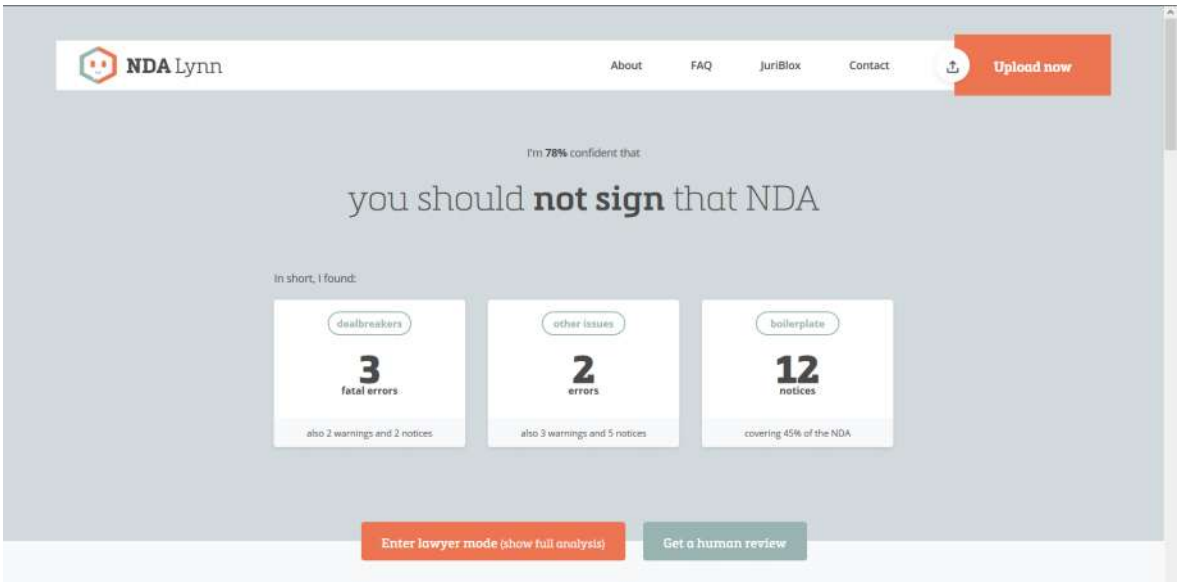

Figure 4. The output of the initial evaluation system, reporting three fatal errors (issues so serious they by themselves imply the NDA cannot be signed), two errors (serious issues for consideration) and 12 notices (informational items).

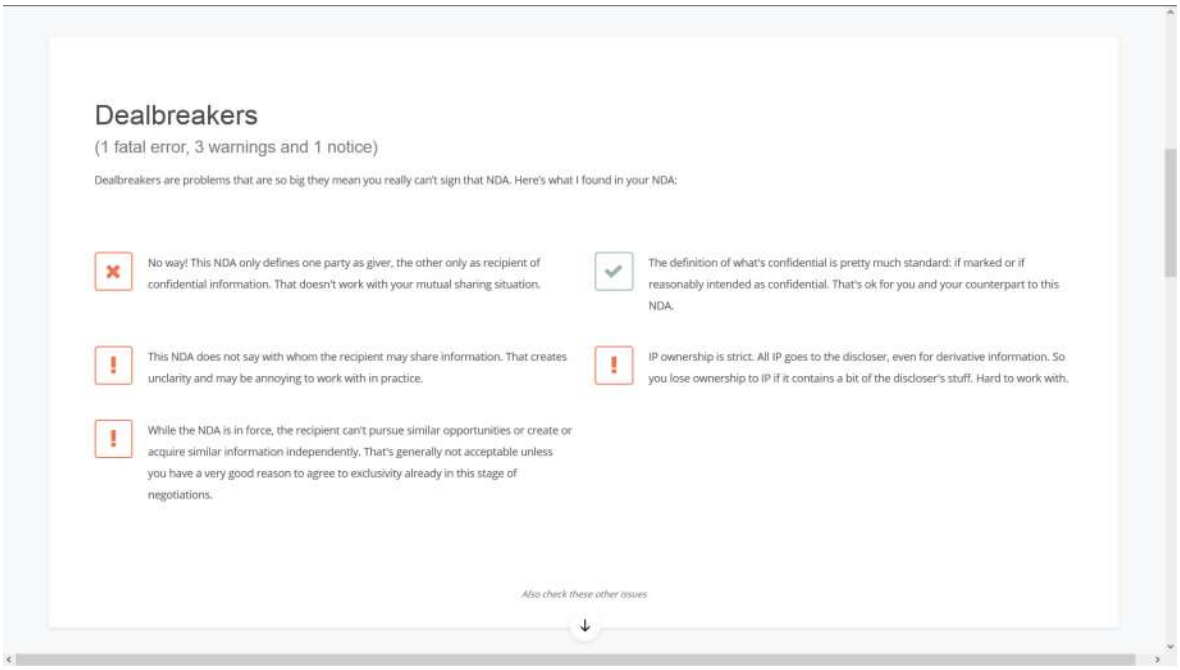

Figure 5. The output of the initial evaluation system for the most serious issues. In this example, one issue is reported as a deal breaker (the NDA is a one-sided NDA, while the use case is that both parties intend to share confidential information). Three issues are reported as worthy of serious consideration (a need-to-know scope is recommended, IP ownership for the discloser is an unusual requirement and an exclusivity period should be carefully considered), and one issue (definition of what constitutes confidential information) is reported as informational only.