CHALMERS
UNIVERSITY OF TECHNOLOGY

Classification of Legal Documents

A Topic Modeling Approach

Master's thesis in Computer Science - Algorithms, Languages and Logic

HANNA CARLSSON & TOBIAS LINDGREN

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Classification of Legal Documents

A Topic Modeling Approach

HANNA CARLSSON & TOBIAS LINDGREN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Classification of Legal Documents
A Topic Modeling Approach
HANNA CARLSSON & TOBIAS LINDGREN

Supervisor: Moa Johansson, Department of Computer Science and Engineering
Advisor: Olof Heggemann, Founder of Eperoto and Johan Thelin, System Architect at Eperoto
Examiner: Krasimir Angelov, Department of Computer Science end Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Classification of Legal Documents:
A Topic Modeling Approach
HANNA CARLSSON & TOBIAS LINDGREN
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Entering a civil dispute presents financial risks for all parties involved, and sometimes all parties may end up losing money. Eperoto is a legaltech start-up in Gothenburg that aims to solve this problem by providing a tool for risk analysis of outcomes of civil disputes. They want to use information about previous cases to improve their tool further and make better analyses of the current disputes. The category of a dispute could play an essential role in the risks involved in a dispute. It could also be used to make more accurate predictions of a dispute based on statistics from previous disputes of the same category. Manually annotating every case is a very time-consuming and costly task.

In this thesis, we develop and evaluate an unsupervised system based on topic modeling for classifying civil dispute judgments into categories. The system presents similar results to previous similar supervised systems in terms of f1-score. The created system managed to classify 67% of the tested documents correctly.

Overall, the system for categorizing civil disputes performed well, especially considering that it is an unsupervised system. Being able to automatically categorize the disputes with an accuracy of 67% significantly reduces the manual work needed to categorize disputes and contributes to improving Eperoto's tool.

Keywords: machine learning, topic modeling, LDA, text classification, unsupervised, multi-class classification, natural language processing, civil disputes

Acknowledgements

We would first and foremost like to thank our advisors at Eperoto, Olof Heggeman and Johan Thelin, who supported us with their expertise, and provided us with the necessary resources to complete the thesis. We would also like to thank our supervisor, Moa Johansson, for guiding us and providing us with essential and valuable feedback.

Hanna Carlsson & Tobias Lindgren, Gothenburg, June 2021

Contents

List of Figures

List of Tables

1
Introduction

In 2020, the Swedish court received a total number of 212 580 cases, and out of these, 85 367 were civil dispute cases [1]. These cases can range from personal disputes regarding the ownership of a TV stand between two individuals to insurance disputes between companies regarding millions of SEK. One of the main problems with civil disputes is that they are challenging to navigate. Some people get into civil disputes without knowing what the process is like, how long the process might be, and how many possible outcomes there could be. Lengthy processes are often very costly, which in the end might cost more than the plaintiff demanded in the first place. Potentially, this could result in a situation where both parties lose money and time. In this case, both parties would have gained in settling before going to court, or early on in the process. Eperoto, a Gothenburg-based start-up, addresses this problem with a software tool. The software tool aims to provide evaluations of civil disputes in an unbiased and rational way by analyzing the values and risks involved in the dispute. Their tool takes information about the dispute in question as parameters and outputs each identified outcome of the dispute. It also outputs the most probable financial outcome of the case, which can be used as an indicator of how to approach a settlement.

The field of law is centered around text, which means natural language systems have played a role in the sector for a long time. However, in recent years there has been an increasing interest in applying Natural Language Processing (NLP) to a broader range of areas within law [2]. One example of an NLP application within the domain of law is predicting judicial judgments from the European Court of Human Rights [3]. Additional examples are extraction of knowledge from previous Russian court records that could be relevant for a specific case [4], and categorizing legal subject matters in the High Court of Australia [5].

When assessing a civil dispute, it is important to know the category of that dispute. Examples of such categories include the construction industry, residential, and divorce disputes. However, there are no standardized categories of civil disputes. Information about how costs and timeframes of a specific dispute category affected the outcome of a case could be of great importance when analyzing new disputes. A possible method to obtain this data is to use old cases and extract the category, price, costs, and timeframe from them. The focus of this master thesis will be on creating an automated system for classifying civil disputes into categories. The system will be built using topic modeling.

Eperoto's current software tool does not consider the category of the dispute. However, the civil dispute's category plays a significant role in the risks and costs involved. A civil dispute within a particular field or category might be successful to a greater extent or have a higher cost of legal fees than other categories. Currently, Eperoto does not have an automated way of classifying the texts, which is the focus of the thesis. Being able to correctly determine what category previous civil disputes are concerning, combined with other parameters, could make Eperoto's risk analysis of a civil dispute more efficient and accurate. It could become accurate since a more informative decision could be made from data about similar cases (same category). Furthermore, the more efficient dispute resolution tool could result in less expensive and risky disputes for all parties involved.

1.1 Aim

The aim of the master thesis is to create an automated system for classifying civil disputes based on topic modeling which is appropriate for real-world use. The system is built with three modules, a preprocessing module, a topic modeling module, and an annotation module. Unsupervised topic models and semi-supervised models are tested and evaluated against each other. The best performing model is used in the topic modeling module of the final automated system. Primarily, the system should be able to classify the training documents, i.e., the documents available in Eperoto's database. Secondarily, the system should be able to classify new, unseen documents into the identified categories. The system is evaluated using a small subset of human annotated documents measuring recall, precision, and f1-score as well as using known topic model evaluation metrics, including expert evaluation.

1.2 Limitations

The focus of the master thesis was the classification of civil disputes into different categories by comparing and evaluating different unsupervised and semi-supervised approaches of topic modeling to build a system that would automate this task. The data was limited to civil disputes from 40 out of Sweden's 48 district courts, during the period 2015-01-01 to 2020-11-01 and concerned a minimum value of approximately 25 000 SEK, excluding:

- Disputes where the case has been dismissed due to formal reasons
- Where the parties have recalled their claims
- In which one party has lost the case due to inactivity (default judgment)
- Where the parties have reached an agreement outside of court that the court has confirmed in a judgment

The dataset consists of 14 783 cases, which have gone through a trial that has resulted in the court deciding on the dispute outcome.

Since no annotated data was available for the project, we decided to base our system on topic modeling. The thesis focus on two versions of unsupervised topic modeling (LDA, CorEx) and two versions of semi-supervised topic modeling (Seeded LDA,

Anchored CorEx). One topic model was selected and used in a new system that classifies civil disputes.

1.3 Ethical considerations

In the field of machine learning, many ethical considerations need to be taken into account. For topic modeling specifically, it will not be known what the topic models base their decisions on when assigning a document to a topic, except that the decisions are based on the training documents. However, in this case, the model's outcome was not new predictions of an ongoing dispute but rather classifies civil disputes that are already finished. The classification will be used in combination with other assessment criterias and a lawyer's assessment, deciding how to proceed with the dispute.

The outcome of the thesis can potentially be used when making decisions about civil disputes. These disputes could concern large amounts of money, and our system could be used to present similar cases. It is essential to know the margin of errors and limitations when using topic modeling systems and informing the users of the system of them. Using the system to classify documents could result in misclassification. It could be that presenting documents of a category would include additional, misclassified documents that should not belong to that category. Furthermore, it could be that presenting documents of a category would not include all the documents that should belong to that category. If additional misclassified documents were included, the user reading the presented cases would notice the misclassified documents and disregard them. However, if not all documents of a category are presented, a relevant and vital case could be overlooked by the user. Therefore, the recall is more important than the precision for this case.

The data used consists of civil disputes that are all publicly available data requested from Swedish district courts. Even though it is public data, these documents consist of personal information about the plaintiff, the defendant, and the lawyers involved. Some of this information could be considered private for the people involved. Information about particular people will not be presented.

1.4 Outline

Chapter 2, *Theory*, summarises the necessary theory behind the methods and models, which are required to understand the thesis.
Chapter 3, *Related Work*, presents some of the previous work in the field that is similar in terms of domain or techniques to the thesis.
Chapter 4, *Method and Implementation*, describes the data used, the text preprocessing in the system, the method used to find it, and the topic model used in ourd system and method used to find it. Also, the evaluation metrics used to evaluate the different parts of the system and the system itself. Finally, the architecture for the baseline system and our system are presented.

Chapter 5, *Results and Evaluation*, presents the result of our system and the baseline system. An evaluation of the quality of the systems and the different modules of the system are also presented.
Chapter 6, *Discussion*, discusses the results obtained, what could have affected the results, and how our system compares to previous classification systems within the domain. Also, some ideas for future work are presented.
Chapter 7, *Conclusion*, summarises and draws conclusions about the overall best performing model for the task of classifying legal documents.

2
Theory

This chapter introduces the concepts and the research this thesis is based on. It describes topic modeling, different topic models, and evaluation techniques for topic modeling. It also describes the general topic of natural language processing (NLP) and text preprocessing which are essential concepts and techniques used when building topic modeling systems.

2.1 Machine learning

Machine learning is defined as computational methods that use existing information to make predictions or improve performance by optimizing various objectives [6]. Based on the current information available to the model, machine learning models are divided into different learning scenarios. Some of the most common learning scenarios are supervised, unsupervised, and semi-supervised learning [6]. Supervised learning uses an annotated data set and learns by mapping data to its annotated output label. Since it requires annotated data, it is limited to data that is annotated. For many real-world applications of machine learning, annotated data is not available. In contrast to supervised learning, unsupervised learning does not require annotated data. Instead, this type of learning searches for patterns in the data and tries to group or cluster the data accordingly. Since there is no correct output defined, it could be challenging to interpret the results and affect the output. Semi-supervised learning is a combination of supervised and unsupervised learning, as it makes use of a partially annotated data set. It is easier to interpret the results since the data set is partially annotated. It is also possible to use a more extensive data set than for supervised learning since not all data needs to be annotated [6].

2.2 Natural language processing

NLP concerns the interaction between computers and natural languages, specifically how a computer interprets human languages and texts. The different methods for NLP could be divided into three approaches: rule-based methods, statistical models, and artificial neural networks. Statistical models take as input a large set of features (words or documents for NLP tasks) and based on statistical assumptions on this input, the model tries to learn a probabilistic model of the input data. This is done by first extracting information from the words and documents, e.g., word count. Secondly, the model tries to learn patterns in the data using the extracted

information, e.g., word co-occurrence. The model also learns the likelihood of different patterns, forming a probabilistic model of the input data. This is then used to perform various tasks, such as machine translation or text categorization [7]. Topic modeling, described further in Section 2.3, is one example of a statistical NLP approach.

Most NLP models can not use plain text as input directly into the models. This is solved by simplifying the input features into a more basic word representation than plain text, such as bag-of-words or term frequency-inverse document frequency (tf-idf). Bag-of-words is a set of key-value pairs with a word as a key and the number of occurrences of this word as the value. This word representation is effective and easily interpretable for computers. However, the most common words in the text, such as*the, a*, will be considered important since they often occur in the text, while in reality, they might not provide any useful information about the text. Tf-idf, on the other hand, handles this issue by assigning weights to words in the text. Words common in the text will have a smaller weight, while uncommon words will have a larger weight assigned to them. For most NLP tasks, tf-idf performs better than bag-of-words since it takes into account that common words do not necessarily contribute as much to understanding the text as more uncommon words [8, 9]. However, assigning these types of weights can invalidate some statistical assumptions that are needed for statistical models. For example, the bag-of-words assumption, which means each word in a text is independent of each other, is needed for some topic models [10].

2.3 Topic Modeling

Unsupervised topic modeling is a machine learning technique for recognizing topics, or themes, from a set of documents. More specifically, it is a statistical method that analyzes words in documents and attempts to determine the semantic structure of the documents. They use various statistical assumptions, such as the bag-of-words assumption and word co-occurrence measures, to find patterns, and from them, form K topics. The number of topics K must be decided before training the topic model.

Semi-supervised topic modeling is an extension of unsupervised topic modeling. The data set is enhanced with a small annotated part, but with the majority of the data still not annotated [11]. With unsupervised topic modeling, there is no way to control which topics are being generated. However, with semi-supervised topic modeling, one could steer the models to try to generate certain identified topics. This is done by using seed words, which are words that are linked to topics with a specified weight. Thereby, one could choose to pre-assign a word to a topic or use a set of words to define topics beforehand, that the model should steer towards [12].

2.3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA), first presented in the machine learning context by Blei et al. [10], is a generative probabilistic topic model for collections of discrete

data, such as a text corpus [13]. The generative process of the LDA model can be seen below [14]:

1. For each topic $k = \{1, ..., K\}$:
   (a) Draw a word distribution for each topic, $\phi^k \sim Dirichlet(\eta)$
2. For each document $d = \{1, ..., M\}$ in the corpus:
   (a) Draw a document topic distribution, $\theta_d \sim Dirichlet(\alpha)$
   (b) For each word w_i in document d:
      i. Draw a topic from the document topic distribution, $z_i \sim Multinomial(\theta_d)$
      ii. Draw the observed word, $w_i \sim Multinomial(\phi^{(z_i)})$

where K is the number of topics, and M is the number of documents in the corpus. The α and η parameters, and the *Dirichlet* and $Multinomial$ distributions are explained below. First, a topic word distribution is drawn for each topic. Next, for each document, a document topic distribution is drawn. For each word in a document, w_i, a topic is drawn from that document topic distribution, and from that distribution the current word is drawn [14].

For LDA, every document is seen as random mixtures over hidden (latent) topics, where each topic is a probability distribution over words. Each word is said to belong to a topic [10]. An optimal number of topics, $K = \{k_0, ..., k_n\}$, needs to be determined for the model, and each document exhibits all K topics with a probability distribution. Since topic probabilities are expressed as a multinomial distribution, LDA uses its conjugate prior, which is a Dirichlet probability distribution, as its prior distribution. The Dirichlet distribution is a distribution over vectors. The vector values are in the interval $[0, 1]$, and together they all add up to 1. Since Dirichlet is used as the prior distribution, the posterior is also a Dirichlet [10].

As can be seen in Figure 2.1, the only parameters that can be tuned, except for the number of topics, are α and η. α controls the document topics prior distribution, which decides the topics-per-document density. A lower α means that the documents are made up of few topics, while a high α means that documents are made up of a larger number of topics. η controls the topic word prior distribution, which decides the words-per-topic density. This means that with a low η, few words belong to more than one topic, and a high η indicates that most of the words in the corpus make up each topic. Both α and η remain the same for all documents and words in the corpus [10].

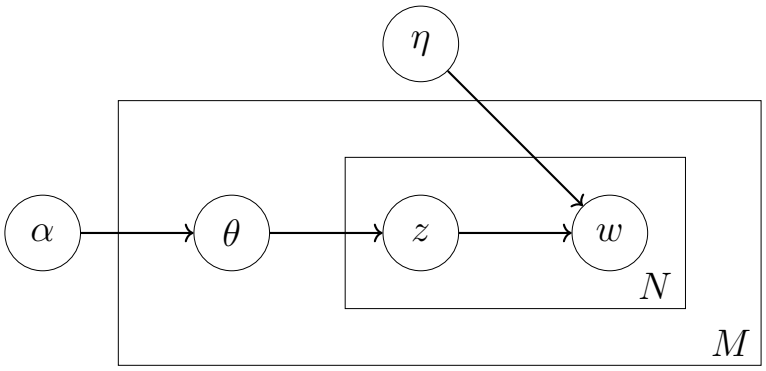

Figure 2.1: Graphical representation of LDA. α and η are the input parameters, θ is drawn from a Dirichlet distribution given α on a document level, z is a topic and w is a word. The inner box represents a document which contains N words, and the outer box represents all the M documents. Let $m \in M$ and $n \in N$, then z_{mn} is the topic for the n-th word in the m-th document, w_{mn}.

In the LDA model, words are not directly added to a topic, but rather a probability is calculated for a word belonging to that topic, similar to that of topics to documents [13]. LDA is a well-established topic model and has previously been applied to several different domains [15, 16].

An intuitive example of how LDA works is presented using the documents 0, 1, and 2 in Table 2.1. A document is made up of different words, and each word is part of each topic with a probability. For this example the LDA model is given two topics, which can be interpreted as *politics* (blue color) and *sports* (green color) topics. In Table 2.2 the probability that each word belongs to the two topics can be seen. Each document belongs to the topics with a probability, and the probability that documents 0, 1, and 2 belong to topic *politics* and *sports* can be seen in Table 2.1 and is based on the words in the documents.

Document	**Words**	**Topic "Politics"**	**Topic "Sports"**
0	Biden was elected president yesterday.	0.963	0.037
1	The L.A. Lakers won the NBA.	0.069	0.931
2	Biden likes to play basketball.	0.479	0.521

Table 2.1: An example of documents where the words in the document are color coded by the topic the words is said to belong to. The assignment of the whole document to the two different topics can be seen in the right-most columns and is a reflection of the words in the document.

	Words								
Topic	*Biden*	*elected*	*president*	*L.A.*	*won*	*NBA*	*play*	*basketball*	...
"Politics"	0.150	0.190	0.280	0.110	0.005	0.001	0.001	0.001	...
"Sports"	0.005	0.001	0.090	0.120	0.250	0.210	0.190	0.130	...

Table 2.2: An example of how words in a corpus are assigned a probability for belonging to each topic. This probability is seen as an representation of which topic a word belongs to.

2.3.2 CorEx

CorEx is an alternative approach to topic modeling through correlation explanation. In contrast to LDA, CorEx does not make generative assumptions of topic structure but rather tries to identify *maximally informative* topics. Ultimately, CorEx seeks to maximize the total correlation (TC).

Total correlation is a measurement, used in probability theory, for mutual information among many variables [17]. This is also called multi-variate information or multi information [18]. It is expressed as

$$TC(X_G) = \sum_{i \in G} H(X_i) - H(X_G) = D_{KL}(p(x_G)|| \prod_{i \in G} p(x_i)). \tag{2.1}$$

The last part of the equation expresses the total correlation as a Kullback-Leibler Divergence, D_{KL}. This is a measurement of how one probability distribution is different from another reference probability distribution, also referred to as relative entropy. X_G denotes a sub-collection of n discrete random variable where $G \subseteq \{1, ..., n\}$, the entropy of X is denoted as $H(X)$, and the mutual information between two random variables are given by $I(X_1 : X_2) = H(X_1) + H(X_2) - H(X_1, X_2)$ [11, 18].

In a topic modeling context, Y represents a topic to be learned and X_G represents a group of words. The topic model is interested in grouping multiple groups of words into multiple topics. The latent topics are denoted as $Y_1, ..., Y_m$ and corresponding groups of words X_{G_j} for $j = 1, ..., m$. In order to maximally explain the dependencies of words in documents through latent topics, the CorEx model seeks to maximize the lower bound of this expression [11, 18]:

$$\max_{G_{j, p(y_j|x_{G_j})}} \sum_{j=1}^{m} TC(X_{G_j}; Y_j)$$

. The CorEx model converges when the change in total correlation is smaller than a defined epsilon parameter, that defaults to 1e-5, according to the authors of the CorEx models implementation [11].

There exists a sparsity optimization for CorEx, which is orders of magnitude faster than the regular/naive version of CorEx. The CorEx algorithm that uses the sparsity optimization has a time complexity that is comparable to LDA. In a previous evaluation of document clustering (homogeneity), document classification, and topic

coherence, CorEx outperformed LDA in all categories for two different data sets [11]. LDA relies on count data which contains more information than binary data, which CorEx relies on. Still, for a disaster relief article data set and a clinical note data set CorEx performed better than LDA. However, the effect could become more noticeable as document size grows since the data sets used in previous evaluations have been relatively small [11].

2.3.3 Seeded LDA

Seeded LDA is an extension of LDA, with the addition of seed words. Seed words are words that are pre-assigned to a specific topic. These seed words are used to extend the multinomial topic-word probability distribution to be a mixture of two multinomial distributions. One is a seed word-topic distribution, and one is a regular distribution. The document-topic distribution is also extended to steer the model to choose document-level topics based on the existence of seed words in the document. A strength value is associated with each seed word, which controls the certainty of that seed word belonging to the pre-assigned topic compared to the words that are not seed words. Experiments presented by Jagarlamudi et al. show that Seeded LDA performs better than LDA according to f1-score and variational information between clusters [19].

2.3.4 Anchored CorEx

Anchored correlation explanation is based on the CorEx algorithm but also makes use of seed words, called anchor words in this context. By constraining the optimization, a word X_i can be anchored to a specific topic Y_j. An anchor strength is assigned to all the anchor words, which indicates the certainty that the anchor words belongs to their pre-assigned topics. This algorithm allows multiple words to be anchored to one topic and for a single word to be anchored to several topics. Through the use of anchor words, one can steer the topic model towards less dominant themes. In previous tests, when tested against CorEx, anchored CorEx performed better than CorEx regarding homogeneity (document clustering) and adjusted mutual information (similarity between clusters) [11]. However, it did not improve or negatively affect the topic coherence compared to the unsupervised CorEx. These tests were done with two different data sets, one containing disaster relief articles and one containing different newsgroups. The previous tests, tested CorEx and LDA against anchored CorEx and two other semi-supervised topic models, and measured homogeneity, adjusted mutual information, and coherence [11].

2.4 Text preprocessing techniques

Text preprocessing is the conversion from a raw text corpus to well-defined input data. In any NLP system, text preprocessing is an essential part since it defines what is passed to the proceeding steps of the system (training and evaluation) [20]. Many different techniques can be applied in the preprocessing of a text corpus, and some of these techniques, relevant for the thesis, are described below.

2.4.1 Stop words

Stop words are a collection of words that are removed before or after the processing of texts. These words are deemed unlikely to convey any information about the topics and are therefore removed [21]. The removal of stop words is a frequently used text preprocessing technique for topic models, and it is proven to have a positive effect on the results for several different data sets [21]. Usually, the first stop words removed are punctuation, special characters, and numbers since they are often considered uninformative. However, there are cases where these could be considered as valid text. An example might be if the corpus is made up of tweets, then hashtags would be considered informative and should therefore not be removed [21]. Commonly, the most frequent words in a language are also used as stop words (example in English: *'it', 'and', 'the'*). Additionally, they could include domain-specific words (example in the English legal sector: *'plaintiff', 'court'*). It is also possible to use the most common words for the specific corpus as stop words. These words could be identified based on the frequencies and distribution of the words among the documents in the corpus. Additionally, using this approach, words that appear seldom could also be added to the stop words list [5]. The construction of a stop words list is difficult and could affect the resulting model both positively and negatively [21].

2.4.2 Stemming and lemmatisation

Stemming is the process of stripping words to their most basic form (stem) by cutting off the end of the word. There are different techniques to know how much of the word to cut to get to the stem, and they differ between natural languages [21]. However, there is always the possibility of over/under stemming. Over stemming refers to cutting off too much of a word, and under stemming refers to cutting off too little, both of which are problematic. An example of stemming is the word *defendants*, which after stemming would become *defendant*.

Lemmatization is the process of replacing a word with its root, known as the lemma. A lemmatization algorithm has knowledge about the roots of words. Therefore, it can reduce the word to its lemma, while a stemming algorithm does not have any information about the word. For example, the lemmatisation of *better* would be *good*, while stemming could reduce it to *bet*, *bett* or *better*. Lemmatization and stemming are both text normalizing techniques that could be used to reduce vocabulary size and increase the matches of words [21].

2.4.3 N-grams

The inclusion of n-grams in a corpus means that words that co-occur often will also be included in the word dictionary. For example, *property defect* would become the bigram *property_defect* and would be interpreted as one word. The most common n-grams to include are bigrams and trigrams (2- and 3-grams). N-grams would be constructed from the corpus, based on how many times the words co-occur. One could specify that two words that co-occur more than X times in the corpus form

a new word (a bigram). Large n-grams would significantly increase the dictionary size, which is not preferred [21].

2.5 Evaluation of topic models

To evaluate topic models, one can either use the approach of evaluating models against each other or evaluating the quality of the topic model itself. There are different methods for each of these approaches, and some are described below [16, 22].

2.5.1 Topic coherence *cv*

Topic coherence *cv* is a way to evaluate topic models by assessing how coherent the generated topics are. By letting its top N words represent a topic, topic coherence *cv* aims to evaluate if topics are coherent or not. Topics are considered coherent if a large majority, or all words, in the topics are related [23]. Each topic has a topic coherence *cv* score, which represents how coherent that specific topic is. The topic model also has a *cv* score, which is an average over all the topics in the topic model *cv* scores.

Topic coherence *cv* is a measure that was presented by Röder et al. in 2015 [24]. It is a combination of a one-set segmentation of the top words, boolean sliding window, and an indirect confirmation measure based on normalized pointwise mutual information (NPMI) and cosine similarity [24]. How *cv* is calculated can be seen in Figure 2.2, and is explained below.

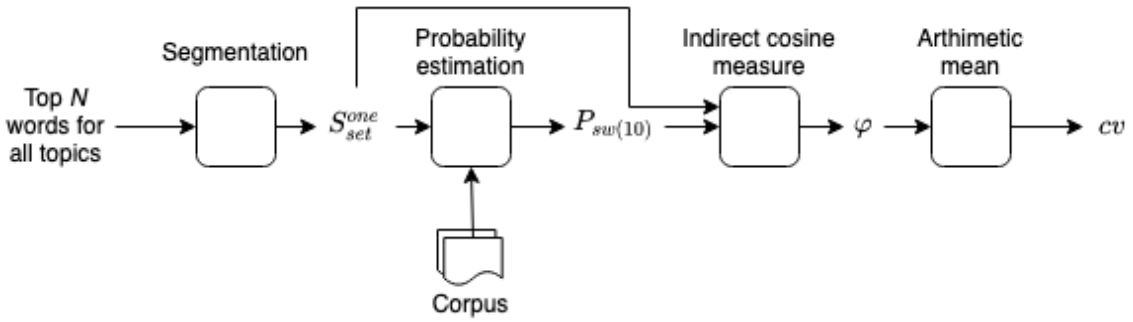

Figure 2.2: Topic coherence *cv* pipeline. It is calculated using the top N words of all topics in a topic model and outputs a *cv* score for the topic model. It is based on four parts; segmentation, probability estimation using boolean sliding window, indirect cosine measure, and an aggregation (arithmetic mean).

The first part of calculating cv is the segmentation. The one-set segmentation creates a set of pairs of each word in the top N words with all the top N words. Let W be the top N words for a topic, and let it be defined as $W = \{W_1, ..., W_N\}$. S^{one}_{set} is the set of word pairs defined as:

$$S^{one}_{set} = \{(W', W^*)|W' = \{w_i\}; w_i \in W; W^* = W\}.$$

An example of the segmentation pair $S_i \in S_{set}^{one}$ with $W = \{sport, basketball, Biden\}$ is $S_i = (W' = sport, W^* = sport, basketball, Biden)$. The S_{set}^{one} that is created is used to estimate the probability and to calculate the indirect cosine measure as seen in Figure 2.2 [24, 23].

The second step is the probability estimation, which is done using boolean sliding window. The boolean sliding window calculates the relevance of words to a documents, and also tries to incorporate word proximity in the calculation. Instead of looking at an entire document when calculating this, the boolean sliding window creates sub-documents of size s ($s = 10$ for cv) when sliding over a documents with a step-size of one word at a time. The boolean document calculation is then calculated for the sub-documents using the number of sub-documents a word or word pair occurs in divided by the total number of sub-documents [24, 23]. The output is denoted as $P_{sw(10)}$, as seen in Figure 2.2. An example of the sub-documents when using sliding window size 2 and document $D =$ "$president, elected, champions$" would be $d_1 =$ "$president, elected$" and $d_2 =$ "$elected, champions$".

The third part in calculating the cv score is the confirmation measure, which uses the indirect cosine measure seen in Figure 2.2. For each segmentation pair $S_i = (W', W^*) \in S_{set}^{one}$ and the probabilities from $P_{sw(10)}$, how much W^* supports W' is calculated, based on the similarity of W' and W^* compared to the rest of the words. The similarity is calculated using an indirect cosine measure. This is done by representing W' and W^* as vectors $\vec{v}(W'), \vec{w}(W^*)$ through calculating the NPMI between each word in the two sets and all words in W. The formula for the NPMI between the words w_i and w_j is given by:

$$NPMI(w_i, w_j)^\gamma = \left(\frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i)P(w_j)}}{-\log(P(w_i, w_j) + \epsilon)} \right)^\gamma \tag{2.2}$$

where the probabilities are estimated using the sliding window, $P_{sw(10)}$. γ is used to put more value on high NPMI values and ϵ is a small value added to avoid the logarithm of 0. The vectors are the sum of the NPMI values of all words in the two sets respectively. The cosine similarity is then calculated between the two vectors [24, 23]. The output, denoted by φ is then used to calculate the arithmetic mean which is the cv score for the topic model as seen in Figure 2.2.

Röder et al. compared different topic coherence measures on several tasks and found topic coherence cv to be the measure which corresponded best with human evaluation [24].

2.5.2 Expert evaluation

For topic models that human users interact with directly, generating topics that correlate with human interpretability is especially important. Topics produced by topic models can have a varying degree of human interpretability, and therefore, it is important to evaluate topic models against this measure [25, 26]. The human evaluation can be considered a gold standard since the other measurements are meant

to replicate the behavior of a human annotator [23]. Within social sciences, using human evaluation for topic interpretability is common [25, 27, 28].

The evaluation is made by a person with expert knowledge within the domain of the task. This person inspects the top words of each topic generated by a topic model and evaluates if the collection of words can be interpreted as a topic in the domain. A drawback of this evaluation technique is that it is a manual task and is often more time-consuming than the automatic techniques [25, 26]. Newman et al. used a rating system with a three-point scale, which was then used by Lau et al. and Röder et al. [24, 25, 26]. The scale goes from 1-3, where Newman et al. called 1 (useless) and 3 (useful), while Lau et al. referred to them as bad (1), neutral (2), and good (3) [25, 26].

3
Related Work

Topic modeling has been used successfully in various fields and for many different purposes. A popular application of topic modeling is clustering of user reviews, where the model categorizes reviews according to their sentiment, positive or negative [29, 30]. Applying topic models to news articles is also a popular application. Newman and Block present topic modeling applied to 18th-century news articles to categorize and find what type of news was present in these historical newspapers and how they changed over time [31]. Lukins et al. used topic modeling for bug localization, where a topic model is trained on the source code of a project, and when given a bug report, can identify where in the code the bug is present [32].

Machine learning is not new within the domain of law. For example, Zhong et al. constructed a machine learning architecture for predicting legal judgments for Chinese criminal cases [33]. They meant a legal judgment is based on several subtasks, and to make an accurate prediction, the results from each of these subtasks are needed. These subtasks include evidence description, information retrieval from law articles, and the prediction of sentences. To solve this, one machine learning model is created for each of the subtasks, and a directed acyclic graph is created where the output of one model is used by another model. Another example of NLP and machine learning in the law domain was presented by Aletras et al. (2016) [3]. They built a model that could predict the outcome of cases in the European Court of Human Rights. It managed to predict wheather or not a case violated the article of the convention of human rights with an accuracy of 79%. Their analysis indicated that the formal facts of a case were the most important predicative factors [3]. Metsker et al. had a different approach to applying NLP to the field of law. They constructed a system for information retrieval of electronic records from court decisions in the Russian court. This system was able to identify the facts that lead to the judgment in certain types of legal cases [4].

Carter et al. made use of topic modeling to categorize the judgments from the Australian high court [5]. Their corpus consisted of 7476 judgments made by the high court from 1903-2015, and an LDA model was used as the topic model. The python library gensim was used to preprocess the text and implement and evaluate the model. The primary aim was to develop the topic model itself, test the appropriate number of topics to prove the usefulness of the topic model by comparing it to the judgment of an experienced human interpreter [5]. By analyzing the results produced, they concluded that the topic models contributed to a new unique way of viewing legal subject matter, as well as a view into how the Australian high court

forms and uses legal practices and concepts [5].

Gonçalves and Quaresma aimed to emphasize the importance of text preprocessing to the multi-label text classification problem by using support vector machines (SVMs) to classify legal texts into different categories [34]. One of the data sets consisted of 8 151 decisions from the Portuguese Attorney General's Office, where all the data was manually classified to a category within the legal scholar. The results from the legal texts dataset showed that removing non-relevant words (pronouns, adverbs, prepositions, etc.), lemmization, term-weighting, and feature subset selection increased the average recall and f1 score with up to 0.1. However, the precision score did not increase compared to only removing special characters as preprocessing. The SVM with the best combination of text preprocessing techniques got a precision score of 0.717, recall of 0.632, and a f1-score of 0.667 [34].

Howe et al., used 6 227 judgments in English from the Singapore supreme court, and aimed to use multi-label classification of legal areas to evaluate different supervised machine learning techniques [35]. All judgments had been manually annotated by experts with the legal area they were within, and therefore, Howe et al. could utilize both supervised and unsupervised learning. About 280 different legal areas had been identified, but they limited the work to only the 30 most common legal areas. Examples of legal areas used for classification are *contract law* and *criminal law*. Among the evaluated techniques were topic modeling. They used Latent Semantic Analysis (LSA) as the topic model and used the output of the LSA as input to a linear support vector classifier (linSVC), which made the classification. They found that this technique was the most accurate of the tested techniques, reaching a f1-score of 0.63 [35].

4

Method and Implementation

This chapter describes the system we have created and its different modules. The data used in the thesis is described in detail in Section 4.1 and the libraries and packages used are described in Section 4.2. The metrics used to evaluate the quality of the system are presented in Section 4.3. The final architecture for our system that classifies civil disputes is described in Section 4.4, and the method for finding how the different modules should be constructed is presented in Sections 4.4.1-4.4.3. A baseline system was created for comparison and is described in Section 4.5.

4.1 Data

The data used were judgments from Swedish district courts. In Sweden, all judgments are public documents. Each judgment contains several sections that present information about the dispute, and these can vary between courts and cases. An example of the training and evaluation data can be found in Appendix A.1.1. The disputes were split into three parts described below.

- *Parties* - Contains the information about the parties (plaintiff and defendant) involved in the dispute. It could also contain information about lawyers, agencies, bankruptcy trustees, or similar.
- *Verdict* - Contains the verdict of the case. This information is usually presented in a list of judgments, depending on the claims by the plaintiff.
- *Miscellaneous* - Contains all sections of a case that are not parties or verdicts. This part is where the majority of the text is, and it is the part used when training the system and classifying the disputes.

A total of 14 783 judgments were used for the thesis, and all texts were written in Swedish. 8 798 out of these documents were used when training the topic model, and 199 of the documents not used when training the model were manually annotated. The annotated documents are described further in Section 4.1.3. The documents came from 40 Swedish district courts, where the number of cases from each court varies between 45 from Lycksele district court to 2 177 from Stockholm's district court. The average length and median length of the judgments were 2 321 words and 1 146 words.

4.1.1 Data exploration

Data exploration is an approach for extracting knowledge from and understanding the data without knowing the exact contents or what to look for [36]. The data was explored by training different topic models and examining the result in terms of top N words per topic. Through this, we found that names and cities were often among the top N words. A topic mainly containing cities and names in the top N words is difficult to interpret as a civil dispute category. The names mainly consisted of the names of the two parties involved in the case and names of witnesses and proxies. It was also discovered that common legal terms and abbreviations were prominent in the top words. Lastly, we found that lemmatization did not visibly have any effect on the topic model's top words or the topic coherence, measured by *cv*. These findings reflected the choices of which text preprocessing techniques to try, described further in Section 4.4.1.1.

4.1.2 Categories

Since there is no official definition of different civil dispute categories, it was difficult to know which categories the system should aim to identify. Therefore, the domain expert, Olof Heggeman, identified common civil dispute categories. He identified 14 general groups of categories, with a total of 25 specific categories. Descriptions for these categories can be seen in Table 4.1, and each category is identified by a *category ID*. Each group of categories contained 1-4 specific categories. These categories were not seen as definite but as an overview of the most common categories. According to the expert, it could exist other civil dispute categories as well. Therefore, the *Other* category was added to the list. The categories defined in 4.1 were used as a benchmark for evaluating the system.

General category	**Specific category**	**Category ID**
Family law disputes	Disputes about law lot violation, distribution of inheritance, and interpretation/invalidity of wills	A
	Disputes about division of property in the event of divorce / termination of cohabitation	B
Construction/contract disputes	Consumer related	C
	Commercially related disputes	D
Rental disputes disputes	Dispute over housing (Business-to-Consumer)	E
	Dispute over premise or lease (Business-to-business)	F
	Dispute over land use	G

Continued on next page

Table 4.1 – continued from previous page

General category	Specific category	Category ID
Disputes regarding right of purchase of goods	Purchase of goods between individuals (Customer-to-customer)	H
	Consumer purchase of goods (Business-to-consumer)	I
	Commercial purchase of goods (Business-to-business)	J
	Other commercial contract dispute	K
Insurance disputes	Insurance regarding injury	L
	Other insurance cases	M
Property defect disputes	Property defects (Customer-to-customer)	N
	Property defects (Business-to-business)	O
	Property defects (Business-to-consumer)	P
	Sale of housing	Q
Debt collection disputes	Debt collection disputes	R
Labor disputes	Labor disputes	S
Dispute regarding intellectual property	Trademarks and patent	T
Companies with liquidity problems / bankruptcy	Companies with liquidity problems / bankruptcy	U
Damages and compensation due to violation of rights or violations of EU law	Aviation or neglect	V
Dispute regarding condominium	Dispute regarding private housing cooperatives	W
Corporate dispute over shares and capital	Corporate dispute over shares and capital	X
Others	Others	Y

Table 4.1: Each general category group is presented and described with their specific categories and their descriptions. In total there are 13 general categories and 25 specific categories (A-Y).

4.1.3 Annotated data

A small subset of data was manually annotated with their corresponding category according to the identified categories above (A-Y). This data was used to evaluate the annotations from the system and measures how accurate the system is. Therefore, this data was not used when training the model. In total, 199 documents were

manually annotated to 17 of the 25 identified categories. The distribution of the categories the documents were annotated to can be seen in Figure 4.1. Category R, *dept collection disputes*, represents almost half of all annotated documents. However, even though this distribution is uneven between the categories, according to the expert's estimation it should be representative of the distribution of categories in the data set.

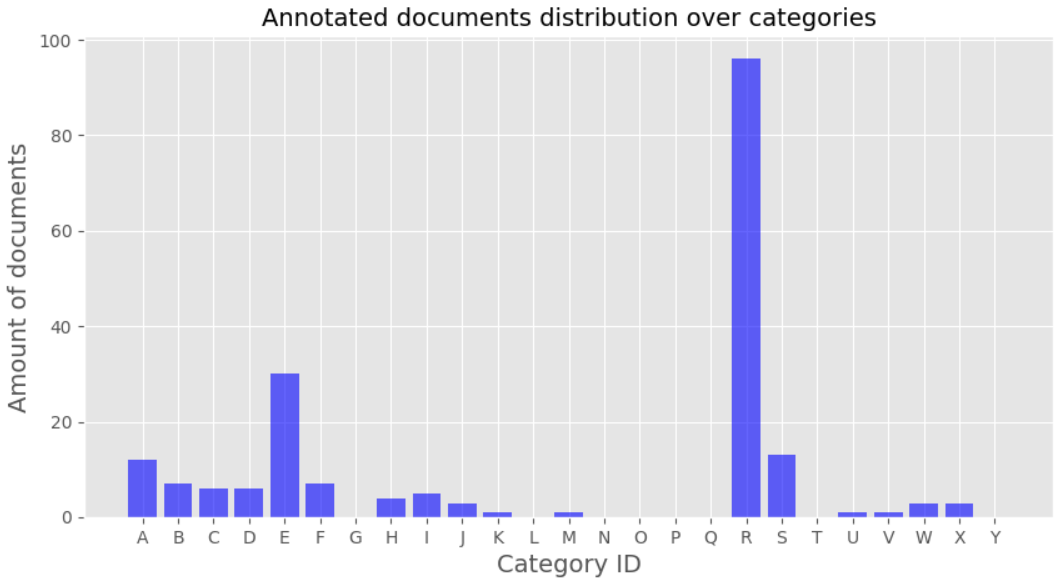

Figure 4.1: The distribution of manually annotated documents to category IDs.

4.2 Libraries and packages

There are various libraries and packages available for the different topic models and evaluation methods. The libraries and packages that were used in the thesis are presented below.

4.2.1 Gensim

Gensim[1] is a free Python topic modeling library that was presented in 2010 by Rehurek et al. and has been cited in academia more than 3000 times [37]. The LdaModel[2] can be used for creating LDA and Seeded LDA models, and it is based on the implementation from Hoffman et al. [14]. The method signatures and the parameters used for the thesis are described below.

LdaModel(*corpus, num_topics, alpha, eta, chunksize, passes, iterations, ...***)**

[1]https://radimrehurek.com/gensim/
[2]https://radimrehurek.com/gensim/models/ldamodel.html

- ***corpus***: This is the training data. It is a matrix of size (number of documents, number of words).
- ***num_topics***: The number of topics the model should have.
- ***alpha***: Controls the α for the LDA model, which controls the document topic prior distribution. It can either be a float value or a predefined value by Gensim defined as:
  - *Single float value*: If a single float value is given, a vector of length equal to the number of topics will be created, filled with the float value.
  - *symmetric*: Sets the value to $1/N$ where N is the number of topics, making each topic equally likely to be part of a document. *Symmetric* is the default value for alpha in the Gensim library.
  - *asymmetric*: Sets the value according to $1/(\text{topic\_index} + \sqrt{N})$, where N is the number of topics and topic_index is the index for each topic, giving each topic a different likelihood to be part of each document.
  - *auto*: Learns an asymmetric prior from the input data by updating the alpha/eta after each chunk of each pass.
- ***eta***: Controls the η for the LDA model, which controls the topic word prior distribution. The eta hyperparameter can be a single float value, a matrix of floats, or a predefined value by Gensim defined as:
  - *None*: This means that no word is assigned to a topic at the beginning of the training. *None* is the default value for eta in the Gensim library.
  - *Single float value*: This means each word is assigned to each category with the same float value.
  - *Matrix (number of topics* $\times$ *number of words)*: The matrix is filled with float values. By increasing the value for a word in a topic vector, that word is more likely to belong to that topic, thereby creating seed words for the model. This converts the LdaModel from an LDA to a Seeded LDA.
  - *symmetric*: For the eta hyperparameter, the value is set to $1/M$ where M is the number of words, which makes each word equally likely to belong to a topic.
  - *auto*: Auto does not calculate one specific value but instead tries to learn an asymmetric prior from the input data by updating the alpha/eta after each chunk of each pass.
- ***chunksize***: Defines how many documents are used at a time in the training algorithm. For our system, this was set to 200.
- ***passes***: The number of times the models are trained on the entire training set, usually called epochs in other machine learning libraries. For our system, this was set to 10.
- ***iterations***: How often a particular loop is repeated for each document. For our system, this was set to 100.

Gensim also provide a coherence model, that can calculate topic coherence *cv*, implemented according to Röder et al. [24]. The values for *cv* range between 0 and 1, where higher values correspond to more coherent topics.

4.2.2 Corex_topic

Corex_topic[3] is a Python package created by the authors of Anchored CorEx, Galagher et al. [11]. It is implemented as described in the paper and explained in detail in the Section 2.3.2. Corex_topic is used to implement both Anchored CorEx and CorEx. By assigning the *anchor_words* hyperparameter with seed words, and the *anchor_strength* hyperparameter to a value above 1, a semi-supervised Anchored CorEx model is created instead of an unsupervised CorEx model. If the hyperparameter anchor_strength is set to value 1 or less, the anchor_words will be ignored.

$$\textbf{\textit{model}} = \textbf{Corex}(\textit{n\_hidden, eps, ...})$$

- ***n_hidden***: The number of topics the model should have. It was tested with different number of topics.
- ***eps***: The epsilon controls when the model converges, by checking if the change in total correlation is less than the epsilon. It was tested with different values.

$$\textbf{\textit{model}}\textbf{.fit\_transform}(\textit{X, anchors, anchor\_strength})$$

- ***X***: This is the training data. It is a matrix of size (number of documents, number of words).
- ***anchors***: The seed words, or *anchor words*, is a list of lists where each list contains the seed words for the topic that is equal to the index of the list.
- ***anchor_strength***: A float value that is assigned to each anchor word. This controls the certainty that the anchor words belong to the topic they are pre-assigned to.

4.3 Evaluation metrics

The system was evaluated using several evaluation metrics. Each metric provided a different measure for the quality of the system. Therefore, combining these metrics provided a more holistic evaluation of our system. A baseline system was also evaluated and compared against our system to give better insight into the performance of our system. The metrics used are described below.

4.3.1 Expert evaluation

The domain expert examined the top ten words of each topic from the topic model used in the baseline and our system. The expert rated each topic on a scale of 1-3 (useless-useful), which is a common way for experts to rate topics from a topic model

[3]https://github.com/gregversteeg/corex_topic

[24, 25, 26]. The topics that the expert could interpret as civil dispute categories were mapped from topic number to the category ID. This mapping was then used in the annotation module. Some topics were mapped to two categories since when examining their top words, it was not possible to distinguish between the two categories. Some topics were not mapped to any categories and were therefore removed from the possible categories to annotate documents to. The expert evaluation was also used when deciding which topic model to use in the system. The decision was based on the average ranking of the topics for each evaluated topic model, described further in Section 4.4.2.1.

The expert used for this thesis to evaluate the topic models was Olof Heggeman. He is the founder of Eperoto, a lawyer since January 2016, and has previous experience working as a judge in training. He was the only expert available for evaluation for this thesis, however, he has extensive experience and knowledge within the domain.

4.3.2 Topic coherence cv

Topic coherence *cv* was used to evaluate the system. Both to evaluate the individual coherence of the topics and the coherence of the topic model. A topic's individual *cv* score can be used to get an overview of the topic quality distribution in the topic model. The average topic coherence *cv* for the topic models was further used to evaluate the different text preprocessing techniques to use in the system, as described in Section 4.4.1.1. It was also used to evaluate which type of topic model and parameters to use in the system, as described in Section 4.4.2.1.

4.3.3 Evaluation on annotated data

To get an indication of how accurate the system is, a small set of documents were manually annotated with one of the civil dispute categories identified by the domain expert that are described in Section 4.1.2. The system's annotations of these documents were compared to the manually annotated categories to evaluate the accuracy of classification. On a model level, the number of correct annotations divided by the total number of annotated documents was of interest, both for the specific and general categories. This was used to get an overview of how well the system performs, both in comparison with humans and in comparison to the baseline. On a category-level, *recall* and *precision* was used to evaluate the systems. For each category c of the identified categories, these metrics look at the number of documents correctly annotated as the category (TP_c), the number of documents incorrectly annotated as the category (FP_c), the number of documents correctly annotated to another category (TN_c), and the number of documents incorrectly annotated to another category (FN_c). The averages across all categories are also calculated. These metrics are further described below.

4.3.3.1 Recall

The recall measure was presented for each category, and it presents the percentage of documents annotated as the category, which should have been annotated to the

category. Another way to describe it is the proportion of documents that were annotated incorrectly to another category, which should have been annotated to this category. Recall is calculated using:

$$recall_c = \frac{TP_c}{TP_c + FN_c}$$

.

4.3.3.2 Precision

The precision measure was also presented for each category. Several documents were annotated to each category by the system. Precision presents the percentage of these documents which were correctly annotated. This is calculated using:

$$precision_c = \frac{TP_c}{TP_c + FP_c}$$

.

4.3.3.3 F1-score

Measuring the accuracy can be difficult from only the recall and precision. F1-score solves this since it is a combination of the two and therefore provides a better measure for the overall accuracy of each category. The f1-score is calculated using:

$$f1\text{-}score_c = 2 * \frac{recall_c * precision_c}{recall_c + precision_c}$$

4.3.4 Probability of system annotation

Each document was annotated with a category from the topic model and the probability of the annotation to that topic. This probability was used in two aspects. It was partly used to examine the probabilities the documents which were annotated correctly had. This measure could indicate which probability range that annotates correctly. It is also used to examine how many documents were annotated with different probabilities by presenting a distribution of these probabilities for our system and the baseline. This gave an overview of the system and how certain it was of its annotations.

4.4 System implementation

The final system architecture is presented in Figure 4.2. As seen in the figure, the system uses topic modeling, which is a statistical model for finding latent topics in documents. These topics are presented as a probability distribution for each document. By identifying what category these latent topics represent, we were able to look at the system as a multi-class classification system. Typical multi-class classification models are supervised machine learning models, where the output labels

are known beforehand and have been trained on annotated data. However, by identifying the topics produced by the topic model, our system functions in the same way.

The system consists of three modules: *text preprocessing*, *topic modeling*, and an *annotation module*. The text preprocessing module is described in Section 4.4.1, the topic modeling module in Section 4.4.2, and the annotation module is described in Section 4.4.3.

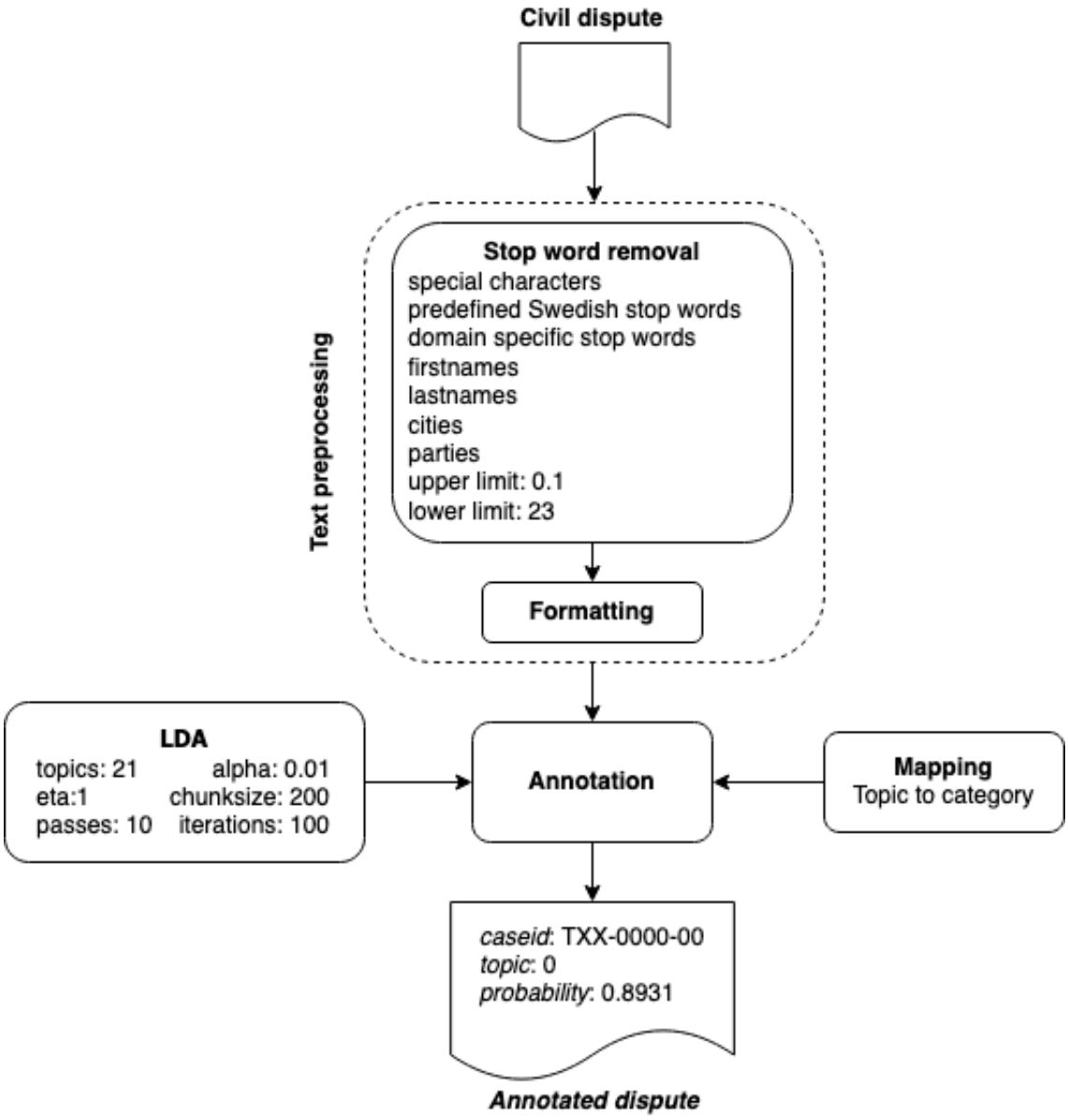

Figure 4.2: Architecture of our system. A single document is used as input and the output is that document with an annotation. The annotation module also needs the trained LDA model as well as the mapping to make the annotation.

4.4.1 Text preprocessing module

The first part of the text preprocessing module is the removal of stop words. In the *stop word removal*, the most common techniques, such as the removal of predefined Swedish stop words and special character was used. Additionally, more specific stop words were used, such as removing domain-specific words, common first names, last names, and cities, as well as the removal of the parties for each case. How these techniques were chosen are explained in the section below.
The *formatting* part converts the preprocessed and filtered documents into a corpus where each document is a bag-of-words. Each word in the documents is represented as a tuple, where the first value is the id of the word and the second is the number of times the word appears in the document. In order to know which word corresponds to which id, a dictionary is created where each word id is mapped to the word.

4.4.1.1 Text preprocessing method

Text preprocessing is a crucial first step in the pipeline of creating and training systems based on topic models and could greatly impact, increase or decrease, the performance of the models [20, 21]. The text preprocessing was divided into three phases; *stop word removal*, *filtering*, and *formatting*. The combination of text preprocessing techniques used when training our system is described in Table 4.2.

	LDA
Stop words	First names, last names, cities, predefined Swedish stop words, domain specific stop words, special characters
Upper limit	0.1
Lower limit	23
Filtering	Remove documents with less than 50 words
Formatting	Bag-of-words corpus and look-up dictionary

Table 4.2: Description of the text preprocessing techniques applied when training the system. The different techniques are applied in the order presented.

The combination of stop words, upper limit, and the lower limit was chosen through a structured grid search approach. Common text preprocessing techniques were applied, but also some less common techniques were applied. These techniques used domain-specific stop words and the extraction of the parties in the case and included these in the stop words for that case. The hypothesis was that removing these words would make it easier to distinguish different topics from each other and make the system more confident in the annotations. All techniques that were tested in the grid search are described below:

- **Stop words**: Words that are removed from the corpus.

  - *Predefined Swedish stop words*: A list of 2400 common Swedish words which are considered to have little to no impact when training NLP models. These are provided by the Natural Language Toolkit (NLTK) corpus package[4], words such as 'är' (is), 'och' (and), and 'vi' (we) are included in this list.
  - *Domain specific stop words*: A list of stop words that are specific for the domain, in this case, the legal domain. These words have been identified together with an expert in the legal domain, Olof Heggeman, and through data exploration. This includes words such as 'käranden' (plaintiff), 'svaranden' (defendant), and 'tingrätt' (district court). It also includes common words such as the weekdays and months, which should not be relevant for the topic, as well as common abbreviations. All the context stop words can be found in Appendix A.1.2.
  - *Names*: A list of common names of people and places. These include the 1000 most common Swedish first names and the 1000 most common Swedish last names, and all the Swedish counties and larger cities. Only Swedish names, cities, and counties are included since the models will be trained on documents from the Swedish judicial system, and a majority of the names and cities in these documents are most likely Swedish. The last names and the first names are taken from Språkbanken, which lists the most common Swedish last names[5], and the most common Swedish first names[6]. The cities and counties are taken from a list of Swedish cities and their county from Wikipedia[7]
  - *Parties*: The names of the parties involved in a civil dispute (plaintiff and defendant) are removed from that specific dispute. The names include individuals and companies. The parties from one dispute are not removed from any other dispute.
  - *Special characters*: removes all special characters from the text which includes all numbers and !"#$%&'()*+,-.:;<=>?@[]^_‘{|} §
- **N-grams**: A function that combines all the words that appear after each other more than 30 times to form new words. The new word would be a combination of the first word underscore the second word (or bigram), for example, 'andra_hand' (subletting).
- **Stemming *(s)***: Stems the words to their basic form, using the NLTK SnowballStemmer[8] for Swedish words.
- **Upper limit**: Defines an upper limit to the number of documents a word can appear in before it is considered to be too general and removed. This limit is expressed as a percentage of the total number of documents. For example, all words that appear in 20% or more documents are removed.
- **Lower limit**: Defines the lower limit to the number of documents a word must appear in. If a word appears in fewer documents than the lower limit,

[4]https://www.nltk.org/api/nltk.corpus.html
[5]https://spraakbanken.gu.se/lb/statistik/lbenamnalf.phtml
[6]https://spraakbanken.gu.se/lb/statistik/lbfnamn.phtml
[7]https://sv.wikipedia.org/wiki/Lista_%C3%B6ver_st%C3%A4der
[8]https://www.nltk.org/_modules/nltk/stem/snowball.html

then the word is considered to be too specific to contribute to a topic. This limit is expressed as a number. For example, all words that appear in less than 15 documents are removed.

The structured grid search tests, used to find the best hyperparameters, were applied to only a subset of the data, 4750 documents. This was the number of documents available at this stage of the thesis. The subset contained documents from 16 different district courts from 2015-2020. All district courts handle all categories of cases, and therefore the subset was considered representative of the entire data set. After applying the different techniques, all documents containing less than 50 words were removed since they were considered too short to be informative.

In order to evaluate the effect of the text preprocessing combinations it needed to be applied to a topic model. Therefore, topic coherence *cv* was used to evaluate the different combinations of techniques against each other. To reduce the number of combinations to test, the grid search was divided into three phases, each testing different techniques, and values. After each phase, the best performing combination of techniques was tested with additional combinations of techniques. The combinations that were tested in each phase, for CorEx and LDA topic models, are presented in Appendix A.2. The combination with the highest overall *cv* score was the combination chosen for that topic model.

After applying the text preprocessing techniques, all the documents that contained less than 50 words were removed from the data set. These documents were considered to be too short and therefore did not provide enough information about the dispute to be able to contribute to the topics of the topic model when training the system. After filtering the data, it was formatted as described above.

4.4.2 Topic modeling module

The topic modeling module is the topic modeling part of the system. An LDA model is used, and the specifications for the LDA model are summarized in Table 4.3 and can be seen in the topic modeling module in Figure 4.2. Since the system's objective was to classify documents, optimally, each document should exhibit one topic with high probability. Also, since there are general categories containing similar specific categories, words should be able to belong to several topics. Therefore, the hypothesis was that low values for alpha and high values for eta were the optimal hyperparameter values and these were also the alpha and eta values used in the final topic model. This model was developed and chosen through extensive tests described below. The LDA model used in the system is already trained. Therefore, the topic modeling module is only passed as input to the annotation module when using the system. The mapping from these topics to the identified categories that were part of the result when training the system is also passed as arguments to the annotation module.

	LDA
Number of topics	21
Alpha	0.01
Eta	1
Chunksize	200
Passes	10
Iterations	100

Table 4.3: The specific hyperparameters for the LDA model used in the topic modeling module.

4.4.2.1 Topic modeling method

To find the optimal topic model, a structured grid search approach was applied. Four different types of topic models were tested. LDA, Seeded LDA, CorEx, and anchored CorEx. These topic models were tested with the optimal text preprocessing found through the text preprocess grid search. The same text preprocessing techniques were used for CorEx and anchored CorEx, as well as LDA and Seeded LDA. The hyperparameters for these models were tuned using grid search, and they were evaluated using topic coherence *cv*. The different hyperparameter values that were tested for each model can be found in Appendix A.3. One of the hyperparameters tested for all models was the number of topics. This was done partly to find the optimal number of topics for an accurate system but also to investigate which categories exist in these disputes. The two semi-supervised topic models, Seeded LDA and anchored LDA were further tested using different seed word dictionaries to more easily be able to generate topics corresponding to the identified categories.

For each type of topic model, three of the best performing combinations of hyperparameters were evaluated by the domain expert. The models that received the highest average topic ranking from the domain expert for each type of topic model were further evaluated on the task of categorizing civil disputes. Each document used when training the model was categorized to a topic, and the probability of these categorizations was examined in combination with *cv* and the top ten words. The Seeded LDA model received the highest average topic ranking, while anchored CorEx had the highest topic coherence.

The results from evaluating the best combination of hyperparameters for the different topic models is summarized in Table 4.4. The LDA model performed significantly better at annotating the majority of the documents with a probability of at least 0.5 and 0.8 compared to the other topic models. It had the second-highest topic coherence *cv* score and second-highest average expert evaluation. A combined assessment of all the results was made to choose which topic model to use in our system. Since the purpose of the system is to classify as many of the documents as possible to identified categories, the LDA model was chosen for this system.

	Cv avg.	Expert evaluation avg.	Probability >0.5	Probability >0.8
LDA	0.718	2.524	**70.1%**	**33.2%**
CorEx	0.704	2.167	26.3%	22.6%
Seeded LDA	0.695	**2.6**	16.3%	0.9%
Anchored CorEx	**0.751**	2.261	25.7%	21.9%

Table 4.4: The results from the different topic models. The best result for each measure is written in bold.

4.4.3 Annotation module

The last module in the system is the annotation module. It takes as input a preprocessed document, the LDA topic model, and the mapping from topics to categories and outputs an annotation of the document. The topic modeling module outputs a probability distribution over the topics. The annotation module uses this probability distribution along with the mapping from topics to categories and classifies the document to the topic with the highest probability in the distribution. If the topic with the highest probability does not have a mapping to a category, the document will instead be classified to the following topic with the highest probability that has a mapping to a category. Each document is classified to one or several categories with the probability of the classification. An example of an annotation can be seen in Figure 4.3.

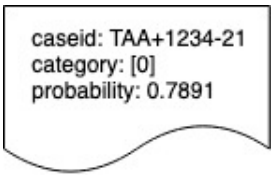

Figure 4.3: An example of the output from the system. The caseid refers to a civil dispute, the category is a list of the categories the document is classified to, and the probability is the probability of the classification from the topic model.

4.4.4 Training the system

The system must be trained before it can be used for classification, and that means the LDA model must be trained. The system was trained using the text preprocessing module, including the filtering part (described in Section 4.4.1.1), and the topic modeling module, as presented in Figure 4.4. The input when training the system were 14 584 documents, but after the filtering phase in the text preprocessing module 8 798 documents were left and used as input into the topic modeling module. Then the topic modeling module was trained with a chunksize of 200, 10 passes, and 100 iterations. The output was the top ten words for each of the 21 topics, which were evaluated by the domain expert and mapped to their corresponding category or categories. The topic model was saved and the mapping from the topics to the

category IDs as well. This was then used as input when running the system as shown previously in Figure 4.2.

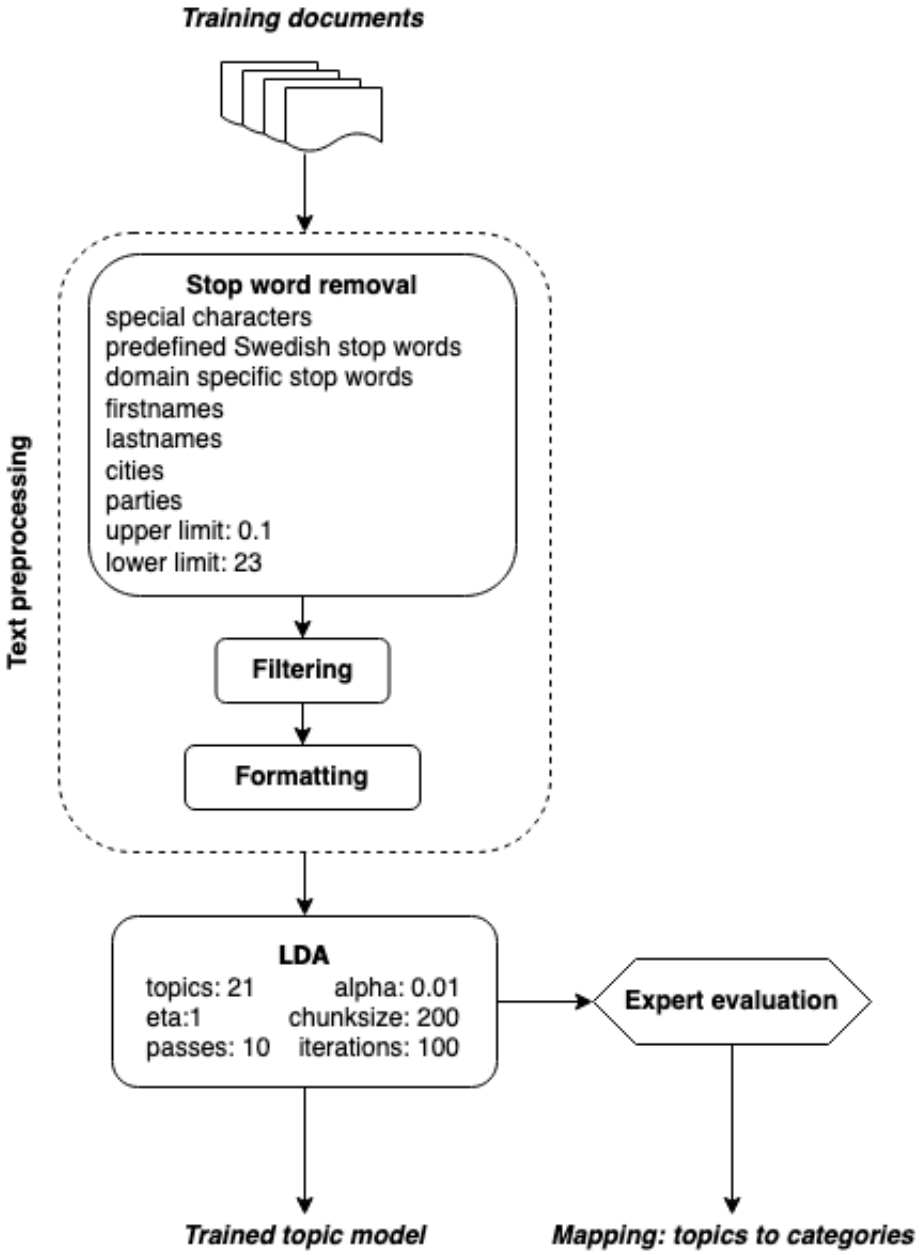

Figure 4.4: Architecture for training the system. A training set is used as input and the output of the training system is a trained topic model, as well as a mapping from topics to categories.

4.5 Baseline classifier

In order to be able to evaluate our system, a baseline system was developed. The baseline is a *default* system compared to our system, and a comparison of the two can be seen in Table 4.5. It is built using the same modules as our system, but

uses default values in each module. In the text preprocessing module, the removal of special characters and most common Swedish stop words are used. In the topic modeling module, an LDA model is used with the default values for alpha and eta (but the same number of topics), and the same annotation module is used, but with another mapping.

As presented in Figure 4.1, around half of the annotated documents were annotated to the dept collection dispute. That means that a baseline model that only classifies disputes as debt collection disputes would perform quite well when evaluated on the annotated disputes. The average f1-score of such a system is actually 0.655, and the average precision is 0.487. This might seem like a good system, but the average recall of such a system is only 0.058, and as mentioned previously, the recall of the system is more important than the precision. The f1-score, is only calculated for the categories that have a precision and recall that is not zero, and both recall and precision can be calculated, which means that for this model, the f1-score is just the f1-score for the dept collection dispute. Also, it would not be a useful model since the system must be able to classify to more than one category. Therefore, the baseline classifier that we present is a system that resembles the final system but with default values, such that it emphasises the effect that tuning hyperparameters and applying different text preprocessing techniques has on the final system. Another advantage of this type of baseline model, is that we additionally get a probability of the classification, which is another metric that we can use to compare our system with.

Module	**Our system**	**Baseline**
Text preprocessing	Special characters, predefined Swedish stop words, *domain specific stop words,* *firstnames,* *lastnames,* *cities,* *upper limit: 10%* *lower limit: 23*	Special characters, predefined Swedish stop words
Topic modeling	topics: 21, alpha: 0.01, eta: 1	topics: 21, alpha: *symmetric (default)*, eta: *None (default)*
Annotation	Annotation using topic to category ID mapping	

Table 4.5: A summary of the differences between our system and the baseline system. The difference lies in the hyperparameter values in the topic modeling module as well as the text preprocessing techniques. These difference are written in italics. The same annotation module is used to be able to compare the results of the two systems, however their mapping from topic's to category IDs differ. The two systems' mapping from topics to category IDs is presented in Chapter 4.

5
Result and Evaluation

This chapter presents the results of our system based on the different metrics described in Section 4.3 and compares it to the results of the baseline system.

5.1 Expert evaluation of topics

To investigate which categories could be identified by the system, expert evaluation was used. It was used to rank the topic from the topic model and map these produced topics to the identified categories if they matched. The expert's ranking of the topic model in our system and the baseline can be found in Table 5.1, where rank 3 is the best rank for a topic and rank 1 is the worst. This gives an overview of the quality of the topics produced by the systems, which in extension, gives an overview of the quality of the systems. There was a significant difference in the average topic ranking and the number of useful and useless topics between the two systems. The baseline system had an average topic ranking of 1.809 while our system had an average of 2.524.

	Average rank	**Rank 1**	**Rank 2**	**Rank 3**
System	2.524	3	4	14
Baseline system	1.809	9	7	5

Table 5.1: An overview of the rankings and average ranking of the topics produced by the topic models for our system and the baseline.

In Table 5.2 the top ten words for each topic in our system's topic model are presented together with each topic's rank and the average probability that the documents annotated as the identified category have. In Section 4.1.2 the expert identified several civil dispute categories. Which category each produced topic corresponds to, according to the expert, is also presented in the table. Almost all categories were identified by examining the top words from the topic model used in the system. Some of the topics exhibit two categories, and those ranked as 1 do not exhibit any topic. *Property defect disputes B2C* (P), *property defect disputes in sale of housing* (Q), and *others* (Y) are the only categories not identified by the system. The topic that had the highest average probability when documents were annotated to it was topic 0, which corresponds to *debt collection disputes* (R). It has an average probability of 0.733, while *rental dispute over premise or lease* (F) (topic 17) had the lowest

average probability of 0.475. Only two categories, category F (topic 17) and U (topic 12) have an average probability of less than 0.5.

Topic	Words	Rank	Identified category	Avg. annotation prob.
0	banken, skuldebrevet, bank, krediten, bankid, konto, skuldebrev, dröjsmålsränta, kredit, borgen	3	R	**0.733**
1	entreprenaden, arbeten, arbetena, bygg, abt, priset, utförda, offerten, hus, utföras	3	C & D	0.567
2	be, ning, an, ningen, ken, uf, kon, ten, livs, der	1	-	-
3	försäkringen, branden, försäkringsfall, trygghansa, försäkring, skadorna, if, inträffat, smyckena, folksam	3	M	0.514
4	artikel, brott, kränkning, försummelse, rättigheter, rättegång, prop, barn, ärendet, uppenbart	2	V	0.695
5	testamentet, dödsboet, testamente, gåvan, gåvobrevet, ärvdabalken, ogiltigt, liv, vilja, avled	3	A	0.668
6	felet, köpet, huset, badrummet, felen, vatten, reklamation, prisavdrag, golvet, köparna	3	N & O	0.625
7	bostadsrätten, bodelningen, gemensamma, köpet, köpeskillingen, fastigheterna, bodelning, gemensamt, försäljningen, egendomen	3	B	0.544
8	kommunen, kommunens, kommun, maskinen, skolan, maskinerna, maskiner, maskin, landstinget, barn	1	-	-
9	mötet, ringde, berättade, minns, mohammed, frågade, hem, bad, saker, pratade	1	-	-
10	lön, anställning, skatteverket, arbetsgivaren, csn, anställningen, arbeta, anställda, las, allmänt	3	S	0.614
11	avtalsbrott, fakturorna, uppdraget, mötet, avtalen, leverans, ingåtts, projektet, punkt, kunder	3	J & K	0.537
12	konto, konkurs, pengarna, konkursboet, medel, konkursbolaget, överföringen, aktierna, skulder, betalningarna	2	U	0.489

Continued on next page

Table 5.2 – continued from previous page

Topic	Words	Rank	Identified category	Avg. annotation prob.
13	väg, mark, bygglov, marken, området, vägen, anläggningen, fastigheter, byggnaden, håkanssons	2	G	0.515
14	olyckan, besvär, hästen, trafikolyckan, inkomstförlust, läkare, besvären, smärta, medicinsk, arbetsoförmåga	3	L	0.676
15	svenska, the, marknaden, of, kunder, upphandlingen, produkter, punkt, anbud, produkten	2	T	0.565
16	föreningen, makarna, föreningens, lägenheten, bostadsrätten, styrelsen, lägenhet, bostadsrätt, medlemmar, brf	3	W	0.503
17	lokalen, hyresavtalet, lokal, lokalerna, hyresgästen, hyresavtal, restaurangen, hyran, lokaler, el	3	F	0.475
18	lägenheten, hyran, lägenhet, hyresavtalet, uppsägningen, hyresrätten, hyresvärden, hyresgästen, jordabalken, förverkad	3	E	0.649
19	bilen, bil, båten, fordonet, fordon, bilens, bilar, köpet, service, körde	3	H & I	0.522
20	styrelsen, aktierna, aktier, miljoner, kapital, aktieägare, styrelseledamot, styrelse, revisor, kapitalet	3	X	0.601

Table 5.2: The topics produced by our system's topic model, its top ten words, the domain expert ranking of the topics, mapping to the identified category IDs, and the average probability to which documents are classified to the category.

In Table 5.3 the same results are presented for the baseline system. Only 13 categories could be identified by examining the top ten words for the baseline system. It is clear that the average probability that each category classified documents with is significantly lower than for our system described earlier. The categories with the highest average probability were *construction/contract disputes* (C and D, topic 16), with an average of 0.236. The category with the lowest average was *rental dispute over premise or lease* (F, topic 0), with an average of 0.094. Our systems classification to category F also had the lowest average probability for the system. However, the average was still 0.475. Category C and D, which had the highest average probability in the baseline system, had an average probability of 0.567 in our system. In our system, all topics of rank two or three could be mapped to identified categories. In the baseline system, all topics of rank three could be mapped, and just some of the topics of rank two could be mapped to a category.

Topic	Words	Rank	Identified category	Avg. annotation prob.
0	lokalen, uppsägning, uppsägningen, hyra, jordabalken, hyresavtalet, lägenheten, ska, hyran, staden	3	F	0.094
1	kr, ska, huset, skador, in, marie, genom, samt, skada, skadan	2	N & O	0.194
2	beslut, kap, enligt, ska, tingsrätten, nämnden, staten, beslutet, genom, björn	1	-	-
3	ska, jan, göran, larsson, santander, kärandena, tingsrätten, dödsboet, christer, parterna	1	-	-
4	bilen, magnus, bil, ägare, fordonet, tingsrätten, bilens, if, försäkringsfall, niclas	2	M	0.173
5	anders, olsson, tomas, staffan, kerstin, kristina, rolf, al, spel, yvonne	1	-	-
6	ska, enligt, även, tingsrätten, fall, år, andra, genom, ersättning, haft	1	-	-
7	lånet, lån, pengar, kr, pengarna, konto, skuldebrevet, parterna, banken, gunnar	3	R	0.156
8	olyckan, besvär, håkan, år, trygghansa, samt, kr, trafikolyckan, ersättning, även	3	L	0.195
9	johan, nilsson, fredrik, michael, daniel, lena, testamentet, inger, kjell, testamente	2	A	0.096
10	avtalet, avtal, parterna, ska, enligt, rätt, punkt, genom, avtalets, avtalen	2	J & K	0.166
11	kr, ska, ersättning, tingsrätten, enligt, målet, betala, belopp, ränta, betalning	1	-	-
12	fel, köpet, felet, köparen, reklamation, köparna, felen, gällande, enligt, säljaren	3	I	0.204
13	ab, ska, talan, bank, målet, betalning, tingsrätten, yrkat, enlighet, sverige	1	-	-

Continued on next page

Table 5.3 – continued from previous page

Topic	Words	Rank	Identified category	Avg. annotation prob.
14	kr, fordran, thomas, enligt, mats, aktierna, ab, konkurs, genom, aktier	1	-	-
15	kommunen, ska, enligt, framgår, mark, kommunens, kap, berg, finns, kommun	2	-	-
16	kr, arbete, arbetet, parterna, enligt, utfört, fel, entreprenaden, arbeten, ska	3	C & D	**0.236**
17	bolaget, ab, bolagets, peter, genom, bolag, uppdrag, haft, även, juni	2	-	-
18	fick, andersson, kom, in, ville, maria, få, eftersom, även, fått	1	-	-
19	fastigheten, lars, johansson, stefan, makarna, ulf, sven, persson, svensson, fastighet	1	-	-
20	föreningen, lägenheten, föreningens, bostadsrätten, lägenhet, kap, tillträde, genom, brf, torbjörn	2	W	0.160

Table 5.3: The topics produced by the baseline system's topic model, its top ten words, the domain expert ranking of the topics, mapping to the identified category IDs, and the average probability to which documents are categorized to the category.

5.2 System accuracy

The system and the baseline were evaluated using annotated data. 199 documents were manually annotated and used in the results and evaluation described below. The annotated data was used to look at the categorization to the specific categories and the general categories. Our system managed to annotate about 67% of the documents correctly for the specific categories, while the baseline managed to annotate 20% of the documents correctly. For the general categories, our system categorizes 69% of the documents correctly, and the baseline system categorizes 36% of the documents correctly.

5.2.1 Recall, precision, and F1-score

The recall, precision, and f1-score for the specific categories are presented in Table 5.4 for our system and the baseline. According to recall, for the categories where it was applicable, eight out of 17 categories in the system had a recall of 0.5 or over,

meaning at least half of all documents were annotated correctly to these categories by the system. The average recall was also 0.5. The precision of the system, however, was not as good. Only nine out of 21 categories had a precision of 0.5 or over. This means that, for only nine categories, at least half of the documents classified to the category were correct. The f1-score presents a general picture of the system's accuracy of the categorizations. For those categories where f1-score was applicable, only four received a score below 0.5. Comparing the average for the different metrics between our system and the baseline, our system performs better for all metrics. Our system performs much better, especially according to f1-score, where the difference between the systems is 0.2.

	Our system			**Baseline**		
Category	**Recall**	**Precision**	**F1-score**	**Recall**	**Precision**	**F1-score**
A	0.67	1.0	0.8	0.58	0.29	0.39
B	0.43	0.6	0.5	0.0	N/A	N/A
C	0.67	0.57	0.62	0.83	0.23	0.36
D	0.17	0.14	0.15	1.0	0.27	0.43
E	0.7	0.81	0.75	0.0	N/A	N/A
F	0.43	0.6	0.5	0.86	0.06	0.11
G	N/A	0.0	N/A	N/A	N/A	N/A
H	0.5	0.67	0.57	0.0	N/A	N/A
I	0.2	0.33	0.25	0.4	0.4	0.4
J	0.33	0.14	0.2	0.67	0.15	0.25
K	1.0	0.14	0.25	1.0	0.08	0.14
L	N/A	0.0	N/A	N/A	0.0	N/A
M	1.0	0.33	0.5	0.0	0.0	N/A
N	N/A	0.0	N/A	N/A	0.0	N/A
O	N/A	0.0	N/A	N/A	0.0	N/A
P	N/A	N/A	N/A	N/A	N/A	N/A
Q	N/A	N/A	N/A	N/A	N/A	N/A
R	0.79	0.84	0.81	0.1	0.77	0.18
S	0.69	0.9	0.78	0.0	N/A	N/A
T	N/A	N/A	N/A	N/A	N/A	N/A
U	0.0	0.0	N/A	0.0	N/A	N/A
V	0.0	0.0	N/A	0.0	N/A	N/A
W	0.33	1.0	0.5	0.67	0.5	0.57
X	0.67	0.4	0.5	0.0	N/A	N/A
Y	N/A	N/A	N/A	N/A	N/A	N/A
Avg	0.50	0.37	0.51	0.36	0.21	0.31

Table 5.4: The accuracy of each specific category for the two systems is presented using three different metrics, *recall*, *precision*, and *f1-score*. The average of each metric is presented as well.

In Table 5.5 the recall, precision, and f1-score are presented for our system and the baseline when categorizing to the general categories. The recall of our system did

not increase significantly from using the specific categories. However, the precision of the model increased by 0.21, which resulted in an overall more accurate system according to f1-score. The accuracy of the baseline was not affected as much as our system. Overall, it was more beneficial for our system to categorize according to the general categories. However, the loss of specificity from not using the specific categories must be taken into account.

	Our system			**Baseline**		
General category	**Recall**	**Precision**	**F1-score**	**Recall**	**Precision**	**F1-score**
AB	0.63	0.92	0.75	0.47	0.38	0.42
CD	0.42	0.71	0.53	0.92	0.5	0.65
EFG	0.7	0.81	0.75	0.84	0.31	0.45
HIJK	0.46	0.6	0.52	0.62	0.44	0.52
LM	1.0	0.17	0.29	0.0	0.0	N/A
NOPQ	N/A	0.0	N/A	N/A	0.0	N/A
R	0.79	0.84	0.81	0.1	0.77	0.18
S	0.69	0.9	0.78	0.0	N/A	N/A
T	N/A	N/A	N/A	N/A	N/A	N/A
U	0.0	N/A	N/A	0.0	N/A	N/A
V	0.0	0.0	N/A	0.0	N/A	N/A
W	0.33	1.0	0.5	0.67	0.5	0.57
X	0.67	0.4	0.5	0.0	N/A	N/A
Y	N/A	N/A	N/A	N/A	N/A	N/A
Avg	0.52	0.58	0.6	0.33	0.36	0.56

Table 5.5: The accuracy of each general category for the two systems is presented using three different metrics, *recall*, *precision*, and *f1-score*. The average of each metric is presented as well.

5.3 Probability of categorization

Each of the annotations made by the system comes with a probability. This section examines what the average probability is for the correct and incorrect annotations of the system and presents the probability distribution of all automatically annotated documents.

5.3.1 Correct and incorrect categorization probabilities

In Table 5.6 the average and median probability for the correctly annotated documents and the incorrectly annotated documents to the specific categories are presented for both our system and the baseline. The results for our system show that the average probability for the correctly annotated documents is higher than for the incorrectly annotated documents. This indicates that the annotations with higher probability are more likely to be correct. Since the median probability is roughly the same as the average, about the same amount of correctly annotated documents

have a probability higher than the average as lower than the average. The same goes for the incorrectly annotated documents. The results for the baseline show the same indications as for our system. However, all values are lower for the baseline.

	Our system		**Baseline**	
	Average prob.	**Median prob.**	**Average prob.**	**Median prob.**
Correctly annotated documents	0.75	0.83	0.21	0.21
Incorrectly annotated documents	0.63	0.62	0.09	0.04

Table 5.6: The average and median probability for correctly and incorrectly annotated documents to specific categories by our system and the baseline.

In Table 5.7, the average and median probabilities for the correct and incorrect annotations of the documents to the general categories are presented. These probabilities are very similar to the probabilities presented in Table 5.6. This indicates that both our system and the baseline are neither more nor less confident in the annotations to the general categories compared to the specific categories.

	Our system		**Baseline**	
	Average prob.	**Median prob.**	**Average prob.**	**Median prob.**
Correctly annotated documents	0.74	0.83	0.22	0.22
Incorrectly annotated documents	0.64	0.62	0.05	0.02

Table 5.7: The average and median probability for correctly and incorrectly annotated documents to general categories by our system and the baseline.

Both the categorization to specific and general categories for the system have similar average probabilities. This indicates that it is probable that annotations with a probability over 0.7 are likely correct categorizations, while it is a bit unclear about the categorizations with probabilities between 0.6-0.7. Categorizations with probabilities below 0.6 are more likely to be incorrect categorizations. However, this is based on data from a very small subset of annotated data, and for the results to be more trustworthy, a more significant annotated data part is needed.

5.3.2 Probability distribution over documents

Since the results above presents the average probabilities of the correctly and incorrectly categorized documents, looking at the probabilities of all categorizations is of

interest. Since high probability categorization were more likely to be correct, looking at the probability distribution over documents could give a further indication of how many documents the system is able to categorize correctly.

In Figure 5.1 the categorization probability distribution over documents is presented. The blue bars represent our system and the green bars represent the baseline system. It is clear that our system is more certain about its categorizations compared to the baseline. Our system can classify most documents, 68%, with a probability over 0.5, and 43% of the documents were classified with a probability of 0.7 or above. The baseline system is less sure about the majority of the documents. 69% of the documents were classified with a probability less than 0.2. The baseline system can not annotate documents to categories that were not identified through its topic model, and the number of low probabilities could reflect that. That means some documents were annotated with the category with the second (or lower) highest probability in the distribution. Since the baseline system did not have as many topics that corresponded with a category as our system, this occurred more often for the baseline. It could be the reason that the baseline system had a much lower probability. It is also possible that the baseline is less certain about its classifications in general.

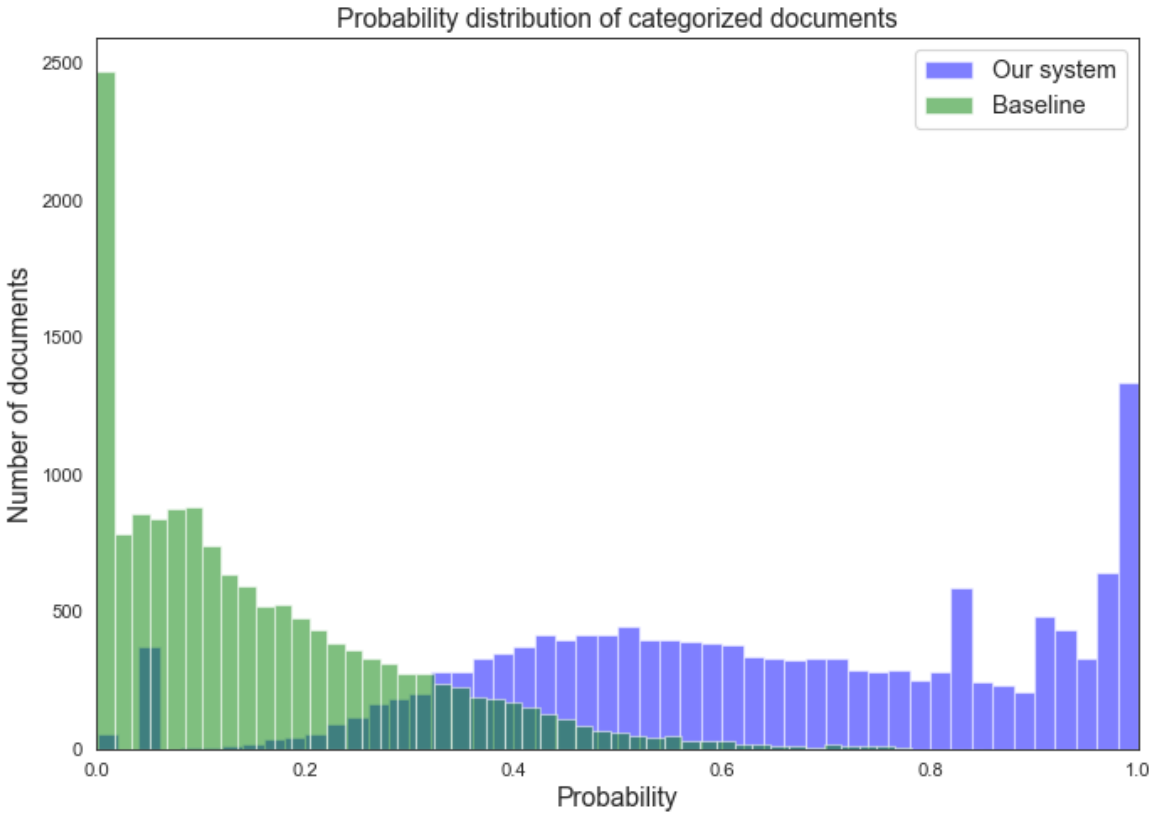

Figure 5.1: Classification probability distribution over documents. Green bars are the baseline system and the blue bars are our system.

5.4 Category annotations

A bubble graph of the annotations is presented in Figure 5.2. This matrix gives a general overview of the annotations and which categories our system classifies correct and incorrect. The manually annotated data set did not have an even distribution between the categories that the documents were annotated to. Nevertheless, most were correct for those documents that were annotated, indicated by the large blue circles in the diagonal. The system instead annotated some documents which were annotated incorrectly to a category of the same general category. For example, the system annotates about the same number of correct documents as incorrect for category *C*, *consumer related construction/contract disputes*. The incorrectly annotated documents are instead categorized as category *D*, *commercially related consumer/contract disputes*. These two are of the same general category and could therefore be difficult to distinguish.

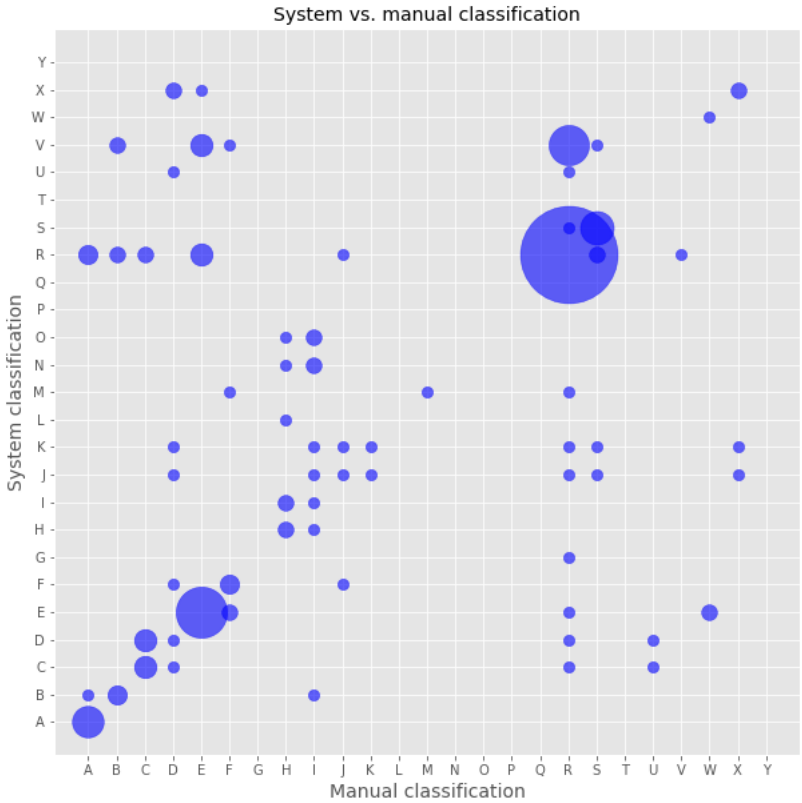

Figure 5.2: The bubble graph for the manually annotated document and the documents annotated by the system. The diagonal squares indicate the documents categorized the same manually and by the system.

5.4.1 Annotations to each category

Which categories our system classified all the documents as, is presented in Figure 5.3. This is presented along with the categorizations of probabilities over 0.3 and 0.7. It is clear from the figure that around one-third of the documents are categorized as *debt collection dispute* (R). This reflects the results presented in the confusion matrix, since more documents were categorized as R and most of them were correct. It is also evident that a large majority of the categorized documents were done so with at least a probability of 0.3. There is no mapping from the systems topic model to three of the identified categories, which is why these categories do not have any documents categorized as them. Category R which has the most documents categorized to it have almost three times as many documents as the second-largest category, *disputes regarding EU-laws* (V). The amount of documents annotated to each category, where there exists a mapping, varies from 210-4996, with a median of 547 and an average of 798 documents.

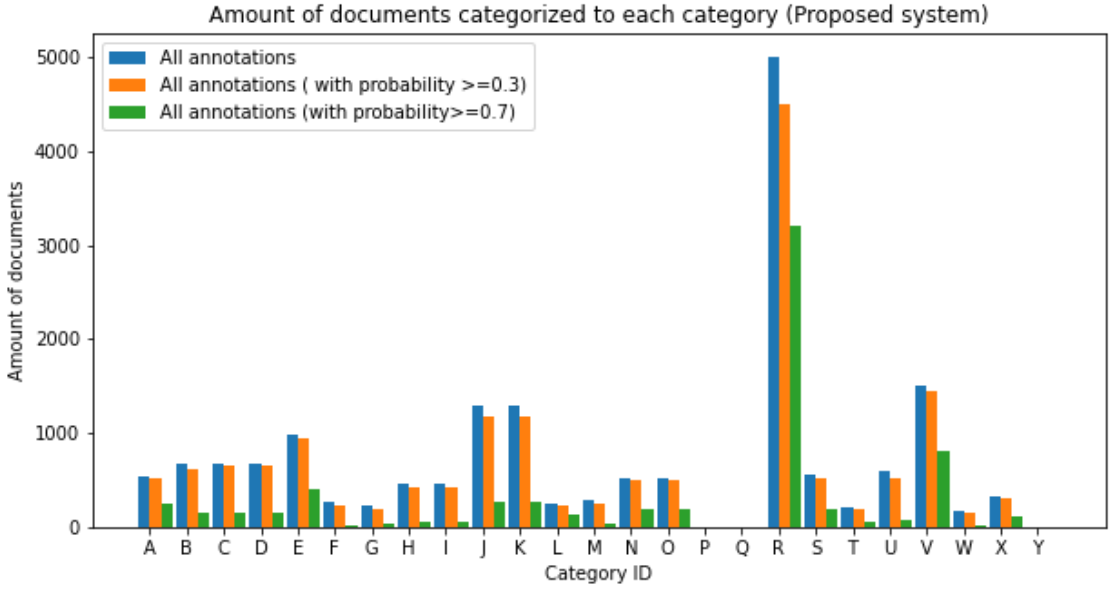

Figure 5.3: The number of documents categorized to each category by our system.

5.5 Topic coherence

Our system and the baseline were also evaluated and compared by looking at the topic coherence from their topic modeling modules. The baseline system had an average *cv* of 0.434 while our system had an average of 0.718. The topic coherence *cv* ranges from 0 to 1, meaning a difference in *cv* by over 0.27 makes our system significantly better than the baseline. The topic coherence for each topic from both topic models can be seen in Figure 5.4. The dashed line represents the average *cv*, the green bars are topics above this average, and red bars are topics below the average. This gives an overview of how coherent the topics are and how this is distributed in the respective models. Comparing the *cv* of the topic model for the baseline and our system, our system clearly presents more coherent topics. Some of

the least coherent topics in our system's topic model are more or equally coherent as the most coherent topics of the baseline topic model. The difference in coherence can be seen by looking at the top words for the two systems. The most coherent topic according to *cv* in our system is topic 5 and the least coherent is topic 8. The top words for these can be seen in 5.2. For the baseline, the most coherent topic is topic 11 and the least coherent is topic 1, and these words can be seen in Table 5.3.

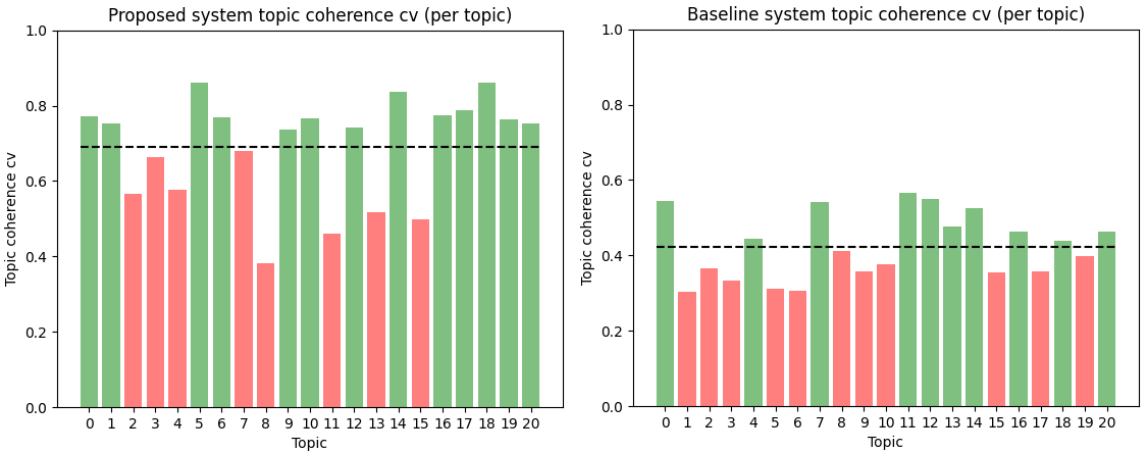

Figure 5.4: Topic coherence *cv* per topic for the topic model used in our system and the baseline. The dashed black line indicates the average *cv* for each model and the green bars are the topics with a *cv* above the average, while the red bars indicate a topic with *cv* below the average.

6
Discussion

Our system manages to partially or fully identify 22 out of the 25 categories identified by the domain expert. When evaluating the system, it classified 67% of the documents correctly. Being able to provide this level of accuracy of automatic classification with no annotations is very beneficial, compared to manually annotating every document. Given these results the system performs well for being an unsupervised system. Of the correct classification the average probability of the classifications was 0.75, while the average classification probability for the incorrect classifications was 0.63. This indicates that the classifications of higher probabilities are more likely to be correct. Also, 43% of all documents are classified with a probability of 0.7 or higher. Given the previous indication, most of these documents would probably be correctly annotated. The classification to the general categories was more accurate than that to the specific categories, in regards to precision and f1-score, and a slight increase in recall and number of correct annotations. Only classifying to the general categories would produce a more accurate model when it comes to annotating correctly. However, the information loss from not having the specific categories is too significant to ignore. Also, by using the specific categories, the system can utilize the general categories, making it possible for a user to filter documents on specific and general categories.

The average recall of the system was 0.5 and the precision was 0.37 for the specific categories. In Section 1.3 we mentioned recall is more important than precision in our system, since missing an important case in a category is considered worse than finding incorrect documents in a category, for this system. This is prioritized in our system, since it is built to classify some documents to two categories instead of only one. This lowered the precision of the system, making that metric somewhat inaccurate, but also increased the recall, which was desirable. Recall, precision, and f1-score are usually considered enough to evaluate a classification system. However, the subset of data that was annotated was very small. Only about 1% of the data was annotated, and that might not be enough data to get results that are representative and can be seen as a generalization of the entire data set. The small number of manually annotated documents was considered throughout the evaluation of the system, and therefore, other metrics for evaluating the system were used to complement. Also by comparing with the baseline results, it is apparent from all the presented results and evaluations that our system performed better than the baseline.

The rest of chapter discusses the different aspects that could have impacted the

construction of the system and the results of the system. Since the thesis aims to build an accurate system for classifying civil disputes, the discussion is centered around the fulfillment of this task and compared to related work, along with ideas for future work.

6.1 Module

This section discuss some more details regarding the results from the system, as well as a more in-depth discussion of the modules of the system. A shorter discussion about the baseline is also included.

6.1.1 Text preprocessing module

The first module of the system was the text preprocessing module. The usage of both domain-specific stop words, parties, and names seemed to increase both the coherence and the interpretability of the topics. Compared to the baseline system's topics' top words, which did not make use of these stop words, they were less interpretable since there were many uninformative words among the top ten words. This is reflected in both the expert ranking of the topics and the number of topics that can not be matched to an existing category. Our system has both higher expert ranking and more topics corresponding to identified categories. Therefore, by looking at the top words generated for each topic, the text preprocessing module can be seen as more beneficial for the aim than the baseline text preprocessing module.

All combinations of text preprocessing techniques and values were not tested since each phase introduced new techniques or values to test, as explained in Section 4.4.1.1. This division could result in an untested combination being overlooked that might had been better for our system than the combination found. However, the grid search was structured to avoid this by testing general values first and adjusting the subsequent phase based on the results from the previous phase. Testing all possible combinations was considered infeasible since it would result in more than 300 000 tests.

When the text preprocessing techniques were applied, documents containing less than 50 words were filtered out and not used in the training set when testing which topic model to use in the system. In the filtering process, around 5 000 documents were removed from the data set of around 14 000 documents. Excluding this number of documents could result in a poorer distribution over categories in the training set. For example, if a specific category of civil disputes often results in short judgments after preprocessing, these would be removed more often than documents of other categories. This removal would lead to a poorer distribution of categories, which would result in a system that does not recognize as many civil dispute categories as it might have if fewer documents were removed. However, the thesis aimed to create an accurate system for classifying the existing documents. Since removing short documents improved the topic coherence of our system, a system with higher

quality was prioritized. In our system for annotating documents, no filtering of documents was done since all civil disputes should be annotated.

6.1.2 Topic modeling module

In order to find the best suited topic model to use in the topic modeling module, a structured grid search approach was applied. This tested four different types of topic models. Topic coherence was used as a filter for excluding models with hyperparameter values that did not produce coherent topics from being further evaluated. Three of the best combinations of hyperparameters according to *cv* were examined further by the domain expert. The use of *cv* as a topic coherence measure was justified since previous research has shown that *cv* was the topic coherence measure that correlated best with expert judgment [24]. Therefore, *cv* was the only metric used to evaluate topic coherence. Among the top-performing models according to *cv*, the difference between each model was minimal, sometimes in the millesimal. By limiting the domain expert to only examining three models for each type of topic model, many models with similar *cv* were not analyzed. However, the risk that one of these models would have performed better in the expert evaluation than the ones examined was considered small. Most of the models that had very similar *cv* and the same number of topics also produced the same or very similar topics. Models that produced the same or similar topics would receive the same or similar expert ranking, and therefore only examining one of these models was considered enough. The domain expert did also not examine models with more than one hyperparameter in common. Therefore, even though the expert examined only three models per type of topic model, no two topic models were considered too similar.

Both of the two tested semi-supervised topic models received a higher expert ranking than their unsupervised versions. A reason for this might be that the seed words were more frequently among the top ten words assessed by the domain expert. By setting a higher strength of the seed words belonging to a topic, they were more likely to be in the top words of a topic initially. While this is desirable, it could be misleading since it does not necessarily mean the rest of the words are of the same topic. It also does not mean that the topic models themselves were *good* at categorizing documents to their identified topics, which was reflected in the low probabilities that they categorized documents with. When tested on the task of categorizing documents, both semi-supervised models were more uncertain in their categorization than their respective unsupervised version, especially the Seeded LDA model. The Seeded LDA model only managed to classify roughly 16% of the documents with a probability over 0.5, which is far less than the LDA which classified about 70% of documents with a probability over 0.5.

6.1.3 Annotation module

The last module in the system is the annotation module. Documents with the highest probability of belonging to a topic with no mapping are not presented. Instead, the following topic that is mapped to a category is presented with its probability. It could be argued that documents should be annotated as these topics anyway since it would indicate that they are unsure about the category. It could also be the case that these documents should be annotated as the *Other* category (X), since the topic model is not sure about them. However, placing all documents the system is unsure of in the *other* category does not help the user of our system. It is deemed very unlikely that a user will search for a dispute in the *other* category, since it can contain documents of almost any category. By instead classifying the document to the first topic which has a mapping to a category, some information can be retained. It could be that the general category is correct or even that the specific category is correct. This can increase the recall of the system, both on a specific category level and on a general category level. This mapping could also be wrong and the document does not belong to any of the identified categories. But instead of using the *other* category, or an *unsure* category, the system uses the probability of the annotation as a guideline to how much one should trust the annotation, or if it needs an additional human check of the classification.

6.1.4 Baseline system

The construction of the baseline system and its use to evaluate our system could be questioned. Preferably the baseline system should be a similar previously presented system. However, there are no previous similar unsupervised systems for the classification. In order to be able to compare our system, a similar system needed to be created to use as a baseline. That meant that the baseline system also should use a topic model, and in order to compare the output of the system, the same annotation module must be used. The usage of the LDA topic model was motivated by it being a very common topic model, and it has been used previously to explore categories in legal texts but not to classify civil disputes. Furthermore, the baseline was needed to get a better overview of what our system can achieve and its shortcomings.

In order to make the baseline system simple but functional, only the most common text preprocessing techniques were used. By doing this, the improvement by adding different techniques for our system became more evident. That was also the reason to use the default alpha and eta values for the LDA model.

6.2 Related work

The work of Howe et al. is similar to ours since they tested and evaluated different classifiers on the task of classifying legal judgment. The system that received the best results was a system using an LSA for topic-document distribution and a linSVC for classification. The results of this system in terms of f1-score, recall, and precision were 0.632, 0.623, and 0.834, respectively. The precision of this system

was far superior to ours, but when comparing the f1-scores to the classification to the general categories, our system is almost on par with Howe et al.'s system.

Similar to our system, Gonçalves and Quaresma tested a text classification system using SVMs against different text preprocessing techniques and evaluated the system using precision, recall, and f1-score. The combination that achieved the highest measures had a precision of 0.717, recall of 0.632, and an f1-score of 0.667. These results were better than compared to our system. However, when looking at the recall and f1-score of the general category classification, the scores are more similar than for the specific category classifications. When looking at the classification of documents to the general categories, our system has a precision of 0.58, a recall of 0.52, and an f1-score of 0.6.

The most noticeable difference between the systems of Howe et al. and Goncalves et al. and our system is that our system is unsupervised, while theirs were supervised systems. A supervised system would probably be more accurate than ours for this task, but the fact that our system performs almost on par with these supervised systems on some measures indicates our system works well.

6.3 Evaluation metrics

Since the system can be seen as a multi-class classification system, calculating recall, precision, and f1-score per topic for evaluation is possible. Instead of only being able to discuss and draw conclusions about the entire model, we can investigate the validity of each category the system finds. Even though recall is more important than the other measures, only looking at the recall would not provide enough information to evaluate the system. If all documents were classified as all categories, the recall of such a system would be perfect, but in reality, the system would be useless. Nevertheless, the recall of the system is still essential. Only looking at the f1-score would not be enough either for our system since the measure looks at both precision and recall. A category could get a high f1-score by having a high precision and a lower recall. Therefore, looking at all these measures was important for a thorough evaluation of the system.

Another evaluation metric used to examine the system was to look at the probability of the system's classifications. It is possible that even though the system is confident in its classification, i.e., it was done with a high probability, it does not necessarily mean that the classification is correct. However, the number of documents in each probability range gives an overview of the certainty of the systems and makes it easier to compare the system with the baseline. The evaluation metric was also used in combination with other metrics. When looking at the probabilities of the correctly classified documents, this information could help draw parallels between the annotated data and the remaining data set. This combination gives an insight into how good the system is and if it fulfills the aim of the thesis. Therefore, even though the probability does not necessarily have to correspond to correct classifications, using this metric to evaluate the system still provides useful information

of the system's quality.

6.4 Future work

The thesis constructs a system for classifying civil disputes by a thorough examination of different text preprocessing techniques and topic models and the construction of an annotation module. This is a new system, and some ideas for using the system's output and how to further extend the system are presented below.

- One possible approach for creating a more accurate system is to use the system's output, the annotations, as input to a supervised classification system. The supervised system can also use more advanced word embeddings, which learn associations between words, which could further improve the accuracy for classifying civil disputes. To ensure the supervised model is trained on mostly correctly annotated documents, only documents annotated with a probability over a specific limit should be used. This could, however, require more documents than what was used in this thesis.
- Another possibility would be to extend this system to try to find all the identified categories and perhaps make the system more accurate. We propose three different approaches to doing so. First of all, one could use the system and only look at the general categories. These documents would be used in a second layer of topic modeling with the same number of topics as the number of desirable specific categories. For example, for the general property defect category, there are four specific categories. All documents classified as one of these four would be the input into another topic model with just four topics. The output of this topic model would be mapped to the specific categories and used as input into the annotation module. Another possible way would be to use a simple rule-based approach. By examining the parties of a case, or the sum it is regarding using some simple defined rules, the specific categories for business-to-business or customer-to-customer could be distinguished. Lastly, one could train another topic model on just the documents that were classified to a topic without mapping to an identified category. This might increase the chance that these documents could be mapped to an category that is not recognized in the current system.

7
Conclusion

The thesis aimed to create a system for classifying civil disputes that could be used in a real-world application. We evaluated different techniques for text preprocessing, different types of topic models, different hyperparameters for these models and created a translation from topic model output to the annotation of a document. The system was then evaluated using a combination of evaluation techniques for a complete overview of the system's performance on this task.

Our system fulfills the aim of the thesis and manages to classify civil disputes without any annotated data. The system can find almost every identified category of disputes, and a majority of the documents were classified with a relatively high probability. Compared to similar supervised systems, our system manages to classify with some measures that are almost on par with the supervised systems. The results from the evaluation showed that documents annotated with a high probability were generally more often correctly annotated. The probability of the annotation is therefore crucial to present to the user of the system.

From the results obtained and the evaluations made, it can be concluded that our system can be used in a real-world application for classifying civil disputes. The value gained from using the system which can classify documents with a 67% accuracy is greater than the information lost. Being able only to annotate half of the documents would contribute significantly to reducing the manual work needed. However, it should be used with knowledge about the limitations described above. It is crucial that a user of the system takes the probability of the classification into the calculations when using the system's classifications to retrieve disputes or basing statistics on them.

Bibliography

[1] Domstolsverket, "Domstolsstatistik 2020," 2021. [Online]. Available at `https://www.domstol.se/globalassets/filer/gemensamt-innehall/styrning-och-riktlinjer/statistik/2021/domstolsstatistik-2020.pdf`.

[2] R. Dale, "Law and word order: NLP in legal tech," *Natural Language Engineering*, vol. 25, no. 1, p. 211–217, 2019.

[3] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the european court of human rights: A natural language processing perspective," *PeerJ Computer Science*, vol. 2, p. e93, 2016.

[4] O. Metsker, E. Trofimov, and S. Grechishcheva, "Natural language processing of russian court decisions for digital indicators mapping for oversight process control efficiency: Disobeying a police officer case," in *International Conference on Electronic Governance and Open Society: Challenges in Eurasia*, pp. 295–307, Springer, 2019.

[5] D. J. Carter, J. Brown, and A. Rahmani, "Reading the high court at a distance: Topic modelling the legal subject matter and judicial activity of the high court of australia, 1903-2015," *UNSWLJ*, vol. 39, p. 1300, 2016.

[6] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[7] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*, vol. 2. CRC Press, 2010.

[8] T. H. Nguyen and K. Shirai, "Text classification of technical papers based on text segmentation," in *International Conference on Application of Natural Language to Information Systems*, pp. 278–284, Springer, 2013.

[9] S.-W. Kim and J.-M. Gil, "Research paper classification systems based on tf-idf and lda schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–21, 2019.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993–1022, Jan 2003.

[11] R. J. Gallagher, K. Reing, D. Kale, and G. Ver Steeg, "Anchored correlation explanation: Topic modeling with minimal domain knowledge," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 529–542, 2017.

[12] K. Watanabe and Y. Zhou, "Theory-Driven Analysis of Large Corpora: Semisupervised Topic Classification of the UN Speeches," *Social Science Computer Review*, pp. 1–21, 2020.

[13] D. M. Blei and J. D. Lafferty, "Topic models," *Text mining: classification, clustering, and applications*, vol. 10, no. 71, p. 34, 2009.

[14] M. Hoffman, F. R. Bach, and D. M. Blei, "Online learning for latent dirichlet allocation," in *advances in neural information processing systems*, pp. 856–864, Citeseer, 2010.

[15] A. Schofield, M. Magnusson, L. Thompson, and D. Mimno, "Understanding text pre-processing for latent dirichlet allocation," in *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, vol. 2, pp. 432–436, 2017.

[16] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, *Evaluation Methods for Topic Models*, p. 1105–1112. New York, NY, USA: Association for Computing Machinery, 2009.

[17] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM Journal of research and development*, vol. 4, no. 1, pp. 66–82, 1960.

[18] G. V. Steeg and A. Galstyan, "Discovering structure in high-dimensional data through correlation explanation," *arXiv preprint arXiv:1406.1222*, 2014.

[19] J. Jagarlamudi, H. Daumé III, and R. Udupa, "Incorporating lexical priors into topic models," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 204–213, 2012.

[20] D. D. Palmer, "Text preprocessing.," *Handbook of natural language processing*, vol. 2, pp. 9–30, 2010.

[21] M. Denny and A. Spirling, "Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it," *When It Misleads, and What to Do about It (September 27, 2017)*, 2017.

[22] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 262–272, 2011.

[23] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," in *2017 IEEE International conference on data science and advanced analytics (DSAA)*, pp. 165–174, IEEE, 2017.

[24] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*, pp. 399–408, 2015.

[25] J. H. Lau, D. Newman, and T. Baldwin, "Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 530–539, 2014.

[26] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pp. 100–108, 2010.

[27] D. Ramage, E. Rosen, J. Chuang, C. D. Manning, and D. A. McFarland, "Topic modeling for the social sciences," in *NIPS 2009 workshop on applications for topic models: text and beyond*, vol. 5, p. 27, 2009.

[28] K. Watanabe, "Newsmap: A semi-supervised approach to geographical news classification," *Digital Journalism*, vol. 6, no. 3, pp. 294–309, 2018.

[29] A. C. Calheiros, S. Moro, and P. Rita, "Sentiment classification of consumer-generated online reviews using topic modeling," *Journal of Hospitality Marketing & Management*, vol. 26, no. 7, pp. 675–693, 2017.

[30] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*, pp. 111–120, 2008.

[31] D. J. Newman and S. Block, "Probabilistic topic decomposition of an eighteenth-century american newspaper," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 6, pp. 753–767, 2006.

[32] S. K. Lukins, N. A. Kraft, and L. H. Etzkorn, "Bug localization using latent dirichlet allocation," *Information and Software Technology*, vol. 52, no. 9, pp. 972–990, 2010.

[33] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal judgment prediction via topological learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3540–3549, 2018.

[34] T. Gonçalves and P. Quaresma, "Evaluating preprocessing techniques in a text classification problem," *São Leopoldo, RS, Brasil: SBC-Sociedade Brasileira de Computação*, 2005.

[35] J. S. T. Howe, L. H. Khang, and I. E. Chai, "Legal area classification: A comparative study of text classifiers on singapore supreme court judgments," *arXiv preprint arXiv:1904.06470*, 2019.

[36] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, "Overview of data exploration techniques," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 277–281, 2015.

[37] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*, Citeseer, 2010.

A

Appendix 1

A.1 Data

A.1.1 Raw data


```
{
    "caseid": "TXX+XXX-XX",
    "data": {
        "parties": [
            "Karande\nPerson 1 \nOmbud: Person 2 \nSvarande
            \nPerson 3 \nOmbud: Person 4"
        ],
        "verdict": ["....."],
        "misc": [
            "DOMSKAL", "Utredningen\n Person 1 har aberopat
            skriftlig bevisning i form av kontrollavgift nr
             XXXXXXXX samt syn\nav fotografi fran
            kontrolltillfallet", ".......", .......
        ]
    }
}
```


A.1.2 List of domain specific stop words

- överklagandet
- överklaga
- överklagande
- klaganden
- dom
- domslut
- domar
- domen
- kärande
- käranden
- kärandena
- svarande
- svaranden
- svarandena

- parterna
- parter
- part
- ombud
- ombudet
- advokat
- advokaten
- tingsrätten
- tingsrätt
- mål
- målet
- målen
- hovrätt
- hovrätten
- domstol
- domstolen
- grund
- tvist
- tvisten
- bilaga
- januari
- februari
- mars
- april
- maj
- juni
- juli
- augusti
- september
- oktober
- november
- december
- måndag
- tisdag
- onsdag
- torsdag
- fredag
- lördag
- söndag
- kr
- kl

A.2 Text preprocessing grid search phases

Phase	CorEx	LDA
Phase 1	***Stop words***: predefined Swedish, domain specific, names, parties, special characters. ***2-grams***. ***3-gram***. ***Stemming***	
Phase 2	***Upper limits***: 0.2, 0.4, 0.6, 0.8 ***Lower limits***: 10, 15, 20	
Phase 3	***Upper limits***: 0.1, 0.15, 0.2, 0.25, 0.3 ***Lower limits***: 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23	***Upper limits***: 0.1, 0.15, 0.2, 0.25, 0.3 ***Lower limits***: 3, 4, 5, 6, 7, 8, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23

Table A.1: Overview of the different text preprocessing techniques and values tested in each phase of the structured grid search.

A.3 Tested hyperparameter values for the topic models

A.3.1 CorEx

Number of topics	Epsilon
15	1
17	0.1
19	0.01
21	0.001
23	0.0001
25	1e-5
27	1e-6
29	

Table A.2: CorEx model hyperparameters used in grid search.

A.3.2 LDA

Number of topics	Alpha	Eta
15	1	1
17	0.5	0.5
19	0.1	0.1
21	0.01	0.01
23	symmetric	symmetric
25	asymmetric	auto
27	auto	
29		

Table A.3: LDA model hyperparameters used in grid search.

Number of topics	Alpha	Eta
19	1	1
20	0.5	0.9
21	0.1	0.8
22	0.01	0.7
23	symmetric	0.6
24	asymmetric	0.5
25	auto	
26		
27		
28		
29		

Table A.4: LDA model hyperparameters used in grid search.

A.3.3 Anchored CorEx

Number of topics	Anchor strength	Epsilon
15	(Only for short seed word dictionary)	-
17	1.5	0.1
19	2	0.01
21	2.5	0.001
23	3	0.0001
25	4	1e-5
27	5	1e-6
29	6	

Table A.5: Anchored CorEx model hyperparameters used in the first round of grid search tests.

Number of topics	Anchor strength	Epsilon
19	1.1	0.0001
20	1.3	
21	1.5	
22	1.7	
23	1.9	
24	2.1	
25	2.3	
26	2.5	
27	2.7	
28	2.9	
	3.1	

Table A.6: Anchored CorEx grid search parameters for the second round of tests with the original seed word dictionary.

Number of topics	Anchor strength	Epsilon
19	1.1	0.001
20	1.3	0.0005
21	1.5	0.0001
22	1.7	
23	1.9	
24		
25		
26		

Table A.7: Anchored CorEx grid search parameters for the second round of tests with the short seed word dictionary.

A.3.4 Seeded LDA

Number of topics	Alpha	Strength
15	(Only for short seed word dictionary)	-
17	1	$1e3$
19	0.5	$1e4$
21	0.1	$1e5$
23	0.01	$1e6$
25	symmetric	$1e7$
27	asymmetric	$1e8$
29	auto	

Table A.8: Seeded LDA model hyperparameters used in the first round of grid search tests.

Number of topics	Alpha	Strength
15	Only for short seed words	
16	Only for short seed words	
17	1	1
18	0.5	10
19	0.1	100
20	0.01	
21	symmetric	
22	asymmetric	
23	auto	
24		
25		

Table A.9: Seeded LDA model hyperparameters used in grid search round two.