sciendo

VOLUME 19, N° 1 2025
Journal of Automation, Mobile Robotics and Intelligent Systems

Identification of Keywords for Legal Documents Categories Using SOM

Submitted: 27th April 2024; accepted: 1st November 2024

Paulina Puchalska, Kacper Krzemiński, Maksymilian Lis, Rafał Scherer, Paweł Drozda, Kajetan Komar-Komarowski, Konrad Szałapak, Andrzej Sobecki, Tomasz Zymkowski, Julian Szymański

DOI: 10.14313/jamris-2025-004

Abstract:
This study aims to use the decision-making process to categorize legal documents by identifying keywords characterizing each legal domain class. The study utilizes the Kohonen Self-Organizing Map method and the Global Vectors for Word Representation (GloVe) model to create an efficient document classification system. As a result, a satisfactory classification accuracy of 71.69% was achieved. The article also discusses alternative approaches implemented to improve classification accuracy, such as the use of Named Entity Recognizer (NER) tools and the RoBERTa model, along with a comparison of these approaches' effectiveness. Challenges related to the uneven distribution of categories in the dataset are also mentioned, and potential directions for further research to enhance the classification results of legal documents are presented.

Keywords: *Document classification, RoBERTa, NLP, GloVe, NER, SOM*

1. Introduction

Document classification holds significant relevance across various domains, including finance, law, medicine, public administration and computer science. It stands as a fundamental challenge within natural language processing (NLP), finding wide application in today's information-driven environment. In this broad context, where precise information categorization is crucial, this article focuses on the classification of legal documents into appropriate categories based on an analysis of the most frequently used words. For this study, a dataset of more than 2,700 legal documents was used, divided into nine distinct fields: civil law, administrative law, pharmaceutical law, labor law, medical law, criminal law, international law, tax law and constitutional law. This study aims to create an effective system for automated classification, enabling the precise allocation of documents to their respective legal categories. Additionally, it seeks to visualize the dataset, highlighting the areas responsible for a specific class together with their defining characteristics.

The structure of the article comprises an overview of existing solutions addressing the challenges of document classification, visualization, and feature extraction, along with an analysis of the different approaches and tools used in this field.

The next section details the chosen approach to solving the problem, together with a characterization of the dataset. Following this, results from the implemented solution are presented, alongside descriptions of additional experiments aimed at improving the outcomes. The article concludes with a discussion of achievements to date and outlines plans for future work to further enhance the results.

2. Classification, Visualization and Feature Extraction of Text Documents

In this section, we will discuss the most popular approaches to natural language classification and analysis and present challenges associated with them, along with the proposed solutions.

2.1. Classification

One of the most commonly used algorithms for text classification is Naive Bayes, based on the assumption of conditional independence of features for a given class. In simpler terms, it assumes that the probability of the occurrence of each word is independent of the occurrence of other words in the same document, making the algorithm fast and efficient in classification tasks. However, the authors of [10] point out a serious problem in the parameter estimation that leads to weaker performance compared to other applied text classification algorithms. To address this issue, the authors propose the use of two empirical heuristics: text normalization and feature weighting, which is particularly effective with a small number of training data. A similar problem is addressed by the authors of the publication *Naive Bayes for Text Classification with Unbalanced Classes* [6], who identify the classifier's drawback when working with imbalanced data sets. As a solution, they demonstrate that normalizing the word vector in each class significantly improves the classifier's performance.

Another equally popular method is logistic regression, which models the probability of a document belonging to a certain class using a logistic function. Though it is an effective classification method, the authors of [21] point out limitations related to high-dimensional data. Typically, a sparsity threshold is applied, which removes sparse features, retaining only a subset of the original ones.

However, this method also has limitations, as there is a possibility that some important features will be cut off, introducing bias into any comparisons. In the aforementioned publication [21], to solve this issue, the authors propose the use of regularized logistic regression, which achieves significantly better results than pure logistic regression.

An algorithm that achieves greater efficiency with high-dimensional data than the two previously mentioned methods is a decision tree. This method makes hierarchical decisions based on the analysis of text features, including dividing the data set into subsets, reducing the dimensionality of the data, and making pattern recognition easier. However, despite their simplicity and effectiveness, decision trees face challenges related to overfitting and their lack of consideration of dependencies between variables. To address these problems, the authors of [12] apply techniques such as pruning and post-pruning, and present more advanced algorithms, such as Intelligent Decision Trees (IDA) and C4.5 [12], which achieve better results in the task of automatic text classification.

Support Vector Machines (SVM), applied to determine the optimal hyperplane that maximally separates different text categories, have been found to surpass the efficiency of the aforementioned algorithms. According to the authors of [18], SVM stand out due to their ability to learn independently of data dimensionality, consider all features without prior selection, and handle sparse document vectors. Despite their high effectiveness, SVM may encounter challenges related to imbalanced training data, which, according to the authors of [16], can be addressed through advanced thresholding strategies.

In the field of automatic text classification, more advanced methods such as Recurrent Neural Networks (RNN) are also employed. These methods involve processing sequential information in texts by recursively updating their hidden states based on previous words, enabling them to capture contextual dependencies and infer document classes [19]. However, even such advanced methods cannot solve every problem in this field. Despite their ability to capture temporal and sequential dependencies in text, RNNs face a challenge known as the vanishing gradient, which limits the effectiveness of modeling long-term dependencies.

Compared to RNN, Long Short-Term Memory (LSTM) models stand out for their effective processing of time sequences using memory modules, which enables the modeling of long-term dependencies in sequential data. Despite this improvement, LSTM also face the problem of the vanishing gradient, albeit to a lesser extent than their predecessor. However, thanks to the use of special gates, such as the input gate, forget gate, and output gate, LSTM can better handle long-term dependencies and feature extraction between words and sequences than the traditional RNN approach [22].

In response to this challenge, the authors of [9] propose a new model, SATT-LSTM, which combines the self-attention mechanism with traditional LSTM to improve handling of long sequences in text.

The self-attention mechanism is an innovative technique that enables the model to assign varying weights to individual elements in a sequence based on their contextual importance. It serves as a central element in Transformer models. Thanks to this mechanism, Transformer models can effectively consider long-term dependencies and global contexts in text, offering a significant advantage over traditional architectures such as RNN or LSTM [7]. This positions Transformer models as one of the most effective methods in automatic text classification.

The most popular and advanced methods for text processing are models based on the Transformer architecture, such as BERT and GPT. The Bidirectional Encoder Representations from Transformers (BERT) model distinguishes itself with its ability to analyze contextual information both preceding and following a given word in a sentence. This capability, known as bidirectional self-attention, empowers the model to capture semantic relationships within the text [3, 13].

Conversely, the Generative Pre-trained Transformer (GPT) model, also based on the Transformer architecture, operates as a generative model, allowing it to produce new sequences of text. Its primary strength lies in its ability to predict subsequent words in a sentence based on contextual cues, thus facilitating the generation of coherent and meaningful texts [20]. Both of these models represent significant advancements in natural language processing, enabling more precise parsing and generation of texts across various contexts.

2.2. Feature Extraction

One of the early and straightforward methods for text feature extraction is the Bag-of-Words (BoW) approach. It represents textual documents by treating each document as a set of words, while disregarding information about grammar or word order [18]. Due to its simplicity and lack of information about sentence structure and word order, BoW is most commonly applied to small, uncomplicated texts.

Alongside BoW, another approach is the application of Term Frequency-Inverse Document Frequency (TF-IDF). In this method, weights are assigned to words based on their frequency in a given document and their relevance to the entire text corpus [4]. Higher weight values indicate greater significance for a word. Although TF-IDF is simple and effective, it results in feature vectors with a high number of dimensions, potentially increasing the risk of overfitting the classification model. To mitigate this issue, various dimensionality reduction techniques are often employed.

Among these techniques are Latent Semantic Analysis (LSA) and Linear Discriminant Analysis (LDA), which are commonly employed in natural language processing tasks.

LSA focuses on capturing hidden semantic relationships in data, while LDA, as a probabilistic model, identifies topics present in documents. With the help of these techniques, classification systems can better handle diverse and complex texts, as well as analyze their semantics more efficiently [4].

Another important technique in text analysis is Principal Component Analysis (PCA), which focuses on reducing the dimensionality of text features. PCA involves transforming data into a new set of uncorrelated variables called principal components, thereby facilitating the analysis of high-dimensional text data without losing essential information [15].

More advanced methods include word embeddings, an advanced word representation technique that assigns numerical vectors to words in a space of specified dimensionality. Word embeddings such as Word2Vec, GloVe, or FastText [17] store word semantics in a manner where words with similar meanings are closer to each other in this space [5]. This facilitates the representation of context and semantic relationships between words, thereby making classification systems utilizing these methods more precise in modeling the meaning of textual documents.

When dealing with keyword extraction, several algorithms enhance the effectiveness of classification systems, such as the Rapid Automatic Keyword Extraction (RAKE) method and the more advanced EmbedRank method. RAKE identifies keywords based on the frequency of word occurrence and their co-occurrence with other words in the text. One significant advantage of this algorithm is its domain and language independence, although it may not fully capture the semantic meaning of the extracted keywords [14].

A solution to this problem is the EmbedRank algorithm, which utilizes word embeddings for keyword extraction. Based on selected criteria, such as frequency and semantics, it assigns an appropriate weight to each word. However, with this method, there is a possibility of obtaining redundant keywords, which can decrease their significance. To mitigate this redundancy, the authors of EmbedRank [1] employed the Maximal Marginal Relevance (MMR) strategy, which helps select relevant and diverse keywords while eliminating redundant ones [1].

2.3. Visualization

Text data visualization plays a crucial role in understanding and analyzing text. One popular visualization method is the word cloud, which presents the most frequently occurring words in a document, assigning them a size proportional to their frequency. However, word clouds have limitations, such as the lack of consideration for semantics or relationships between words. As a result, they are most commonly used for statistically summarizing content [8].

Another popular technique is t-Distributed Stochastic Neighbor Embedding (t-SNE), a nonlinear dimensionality reduction method often applied to visualize high-dimensional data in lower dimensions. By mapping data to a lower-dimensional space, it facilitates the analysis and interpretation of the data structure. Furthermore, it enables the observation of relationships between different text fragments, aiding in a more comprehensive understanding of the context of the analyzed documents [11].

In our study, we utilized the Self-Organizing Map (SOM) method for text document visualization. This method involves transforming multidimensional text data into a two-dimensional map space, where documents with similar themes are represented close to each other. By clustering similar documents, the analysis of the structure and relationships between different categories becomes more accessible, and color-coding areas on the map assists in identifying thematic groups [2].

3. Description of the Approach

We aim to create a system to automatically classify documents into the relevant legal categories.

In the first step, all documents were transformed into vector form using the Kohonen Self-Organizing Maps network. To derive the initial vector representation, the Global Vectors for Word Representation (GloVe)[1] model was employed, producing word embeddings in 100-dimensional vectors for words occurring at least three times in the corpus. The size of the Kohonen network was configured to be 20×20, representing a compromise between task complexity and training time. Subsequently, each query (document) was mapped to the area on the Kohonen map (one of 400 available areas) that best matched its features. To visualise the vectors on a two-dimensional plane, a 20×20 grid was created. To prevent point overlap and enhance visualization clarity, each point was assigned a small random value that was added to its coordinates. This way, we were not limited to just 400 discrete points. The result is the chaotic and unstructured map illustrated in Figure 2, where the colors represent the different classes, according to the legend. Notably, the distances between documents in the same class are similar to those between documents in different classes. In future work, we plan to optimize the layout so that objects from the same class are grouped closer together. However, it is worth noting that projecting vectors of length 100 onto a two-dimensional plane may not be the perfect method of representing this data.

3.1. Dataset

We utilized a dataset comprising 2,722 unique legal questions across nine distinct categories of law. Each record in the dataset includes the legal article number corresponding to the category, the complete content of the question, the main legal category, and the legal department associated with the category. An example of a record looks as follows:

114 The substantive issue, whether the housing community can adopt a resolution regarding the purchase of radio-read water meters for individual units covering their costs, substantive, substantive, civil law.

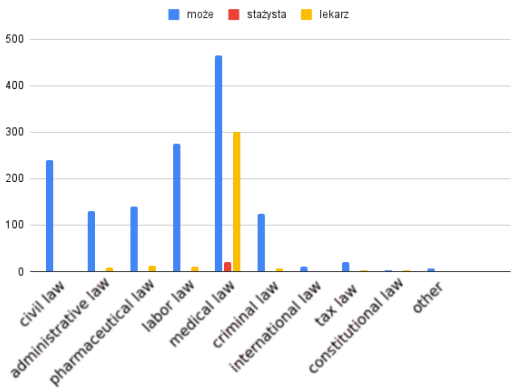

Figure 1. Counts of example words "Może" (maybe), "Stażysta" (intern) and "Lekarz" (doctor) in each category

Specifically, the number 114 corresponds to an article within property law, a subset of civil law.

3.2. Pre-processing Data

To categorize the documents, words uniquely linked to their respective legal-constitutional categories were extracted from the aforementioned dataset. Using the *SpaCy* library (https://spacy.io/) for natural language processing and the *pl_core_news_lg* database containing Polish-language information, all documents from the dataset were analyzed. This process is shown in Table 2 in the *Before* column. Subsequently, 10 parts of speech were removed from the generated dictionary, as their features were deemed irrelevant for document classification. The following parts of speech were removed: *numbering, punctuation, interjection, space, auxiliary, subordinating conjunction, determiner, coordinating conjunction, pronoun, preposition*. The result is shown in *Step 1* column of Table 2. The next step was to eliminate the words that appeared only once in a category, as these were considered too weak in the context of document classification.

In a further step, words that frequently occurred in different categories were removed, as this could significantly reduce the effectiveness of the extracted set of words in the context of model training.

By carrying out an analysis of Figure 1, it is possible to identify three categories of words:

- **"Może" (maybe):**
  - It appears in almost all categories in significant numbers. Similarly ubiquitous words make it harder to distinguish between classes, so such words are removed.
- **"Stażysta" (intern):**
  - This word only appears in one category of document, so its presence increases the probability that a document belongs to a particular class. This is the best possible case for using words for class distinction.

Table 1. Number of words in each category with different thresholds of acceptance for how unique a keyword must be to a category.

Category	Share of words		
	>50%	>90%	100%
Civil law	587	470	466
Administrative law	209	192	191
Pharmaceutical law	253	210	210
Labor law	358	264	263
Medical law	834	599	595
Criminal law	105	90	89
International law	14	12	12
Tax law	42	34	34
Constitutional law	2	2	2
Total	2,404	1,873	1,862

Table 2. Number of words in each category during each step of data preparation

Category	Before	Step 1	Step 2	Result
Civil law	10787	2764	1023	470
Administrative law	5182	1786	562	192
Pharmaceutical law	4742	1489	558	210
Labor law	9707	2081	828	264
Medical law	17503	3447	1407	599
Criminal law	4579	1347	418	90
International law	374	199	40	12
Tax law	1197	468	145	34
Constitutional law	73	40	6	2
Total	54,144	13,621	4,987	1,873

- **"Lekarz" (doctor):**
  - Although it appears in more than one category, the vast majority of its occurrences are attributed to medical law. Removing it from the dataset would adversely affect the quality of the data; as even though it is found in many categories, only one of its occurrences is untraceable. Therefore, such words are assigned to a specific category if 90% or more of its occurrences are in that category.

Table 1 presents how the dataset changes based on the accepted rates of word occurrences. It specifies the minimum percentage of occurrences required for a word to be retained within a given category. An experiment was also conducted with a threshold of 100%, which yielded the number of words the set would contain if all duplicates were removed from all categories. The lower the percentage, the higher the number of words in the selected dataset; however, this often results in lower quality, as many words become less effective in classifying documents as belonging to the appropriate categories. For further work, a threshold of 90% was adopted for the dataset as a compromise between the quantity and quality of words.

The final word count in the dataset during the preparation process is shown in Table 2.

For international law and constitutional law, most of the words extracted from the original dataset were discarded during the elimination process described above. This is partly due to the small number of documents available for these categories in our dataset (with constitutional law comprising 0.15% and international law 0.62% of the dataset).

The TF-IDF measure was used to assess the descriptiveness of the words for each class. This measure was calculated individually for each word in the class, and then the average was calculated to represent the entire class with a single measure. Table 3 displays the results before and after filtering out strong keywords.

3.3. Creating a Training Dataset

The process of creating the training dataset based on the previously-extracted data is outlined below:

1) Extraction of specific sentence fragments from the original dataset to serve as the training set.
2) Creation of a list comprising the most frequently occurring words among the extracted tokens.
3) Iteration through all selected sentences to identify words present in specific categories.
4) Annotation of the occurring words in the appropriate format and assignment of corresponding labels.

This methodology aimed to construct a training set by extracting relevant information from sentences, creating a list of frequently occurring words, and appropriately annotating the data.

3.4. Creating and Training a Model

Using the generated training set, a new network was trained with the *spaCy* library. First, a Named Entity Recognizer (NER) from *spaCy*[2] was used and the prediction labels were defined (e.g.,"civil_law", "administrative_law"). The training data was organized to include the text and entity labels for different legal categories. Each text was transformed into a *Doc* object representing the structure of the document in the given text and label pair. All transformed documents were then collected in an optimized manner in the *DocBin* object.

We can assess the quality of our data visualization by measuring the coherence of the classes in the cluster. To do this, we compute the distances between points from the same class within a certain radius:

$$quality = 1 - \frac{mean(distances)}{threshold} \qquad (1)$$

If all the documents of one class are in the same place, the quality equals 1. If all the points are far from each other by the given radius value, the quality measure will be equal to 0.

The configuration of the NER model in *spaCy* for Polish is based on the *TransitionBasedParser.v2* architecture and token embedding using *MaxoutWindowEncoder.v2*. The training process included training data with NER labels, utilizing the Adam optimizer, and performing model evaluation every 200 steps. Key evaluation metrics focused on the entity-level F1 score.

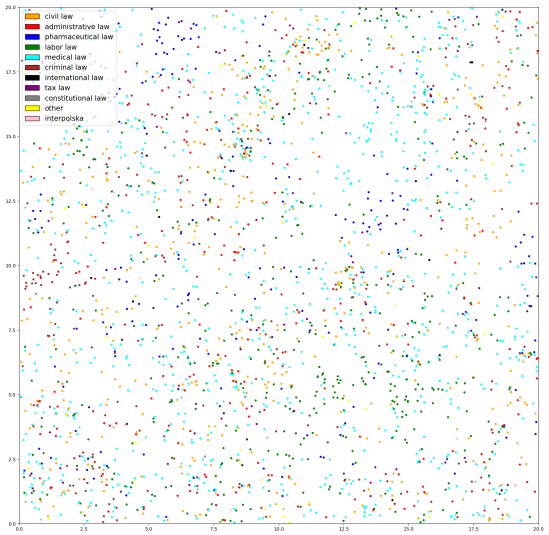

Figure 2. SOM visualization using GloVe embeddings

3.5. Kohonen Network

The final step of the experiment described above was to vectorize all documents contained in the initial dataset using the selected classification. An example of the vectorization performed by the selected algorithm is shown below. For the input sentence: "Can a doctor be employed at the hospital on the basis of an employment contract for a trial period, if they were previously employed at this location as a resident (until the end of 2018)?", the resulting vector takes the form of an array of floating-point numbers:

[0.071..., 0.035..., 0.0, 0.0, 0.071..., 0.0, 0.0, 0.0, 0.0, 0.0]

The array has 10 different characteristics, each defining a different legal category; for the civil, administrative and medical law categories, the values are 0.07, 0.03 and 0.07, respectively, and for each of the other categories, the value is 0.

After converting the documents into vectors, the resulting data was fed into the Kohonen Network. The network's parameters remained consistent with those described previously, where they served as an input stage using GloVe representations. In the initial stage, the self-organizing map was trained on the input data for 1,000 iterations. Then, using the *matplotlib* library, the assignment of "winning" coordinates to each document was presented. To improve the vectorization algorithm used, a weight was added to each word, which was calculated based on the number of occurrences in the document. This means that a word that has occurred, for example, 300 times in a category has a much higher value than one that has only appeared a few times. Once the weights are added to the document vectorization process, the space presented is arranged into sets, as visualized in Figures 2 and 3.

The next step was to create a tag cloud based on the keywords for each of the documents presented in the space above. This began with extracting the words that characterized each document. After distributing these words in the space, the graph took on the shape shown in Figure 4.

Table 3. Number, coherence and descriptiveness of documents before and after using exclusively strong keywords

Category	Before changes			After changes		
	Number of documents	Coherence score	Descriptiveness score	Number of documents	Coherence score	Descriptiveness score
Civil law	863	0.51	0.031	223	0.74	0.041
Administrative law	164	0.85	0.054	84	0.85	0.062
Pharmaceutical law	212	0.79	0.043	101	0.8	0.064
Labor law	327	0.79	0.043	154	0.84	0.055
Medical law	846	0.68	0.023	377	0.71	0.034
Criminal law	192	0.82	0.288	78	0.83	0.325
International law	7	0.94	0.09	3	0.95	0.123
Tax law	61	0.9	0.577	34	0.89	0.550
Constitutional law	2	1	0.038	2	1.00	0.066
Sum	4,402			1,056		

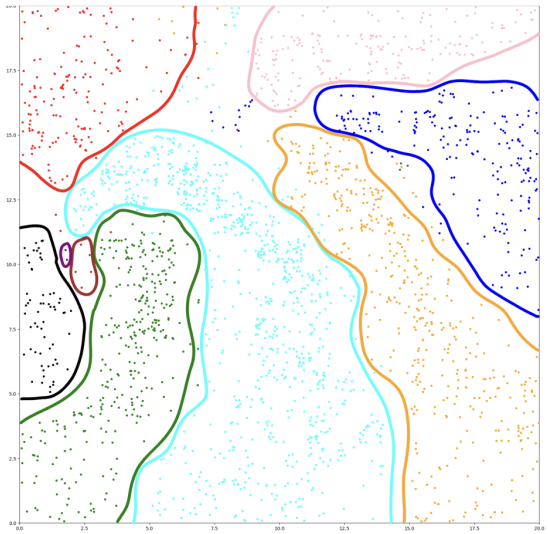

Figure 3. SOM visualization of decision borders, built using NER model

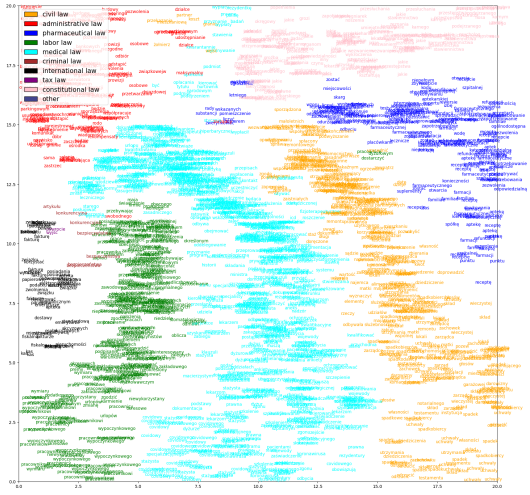

Figure 4. All keywords displayed on SOM

However, by automatically selecting the first word from the classification, when a document was classified as belonging to employment law due to the occurrence of five words from this category, only the first word of the series was taken into the tag cloud. To improve the result, the strongest word from each document's set of keywords was selected, and then duplicates were eliminated.

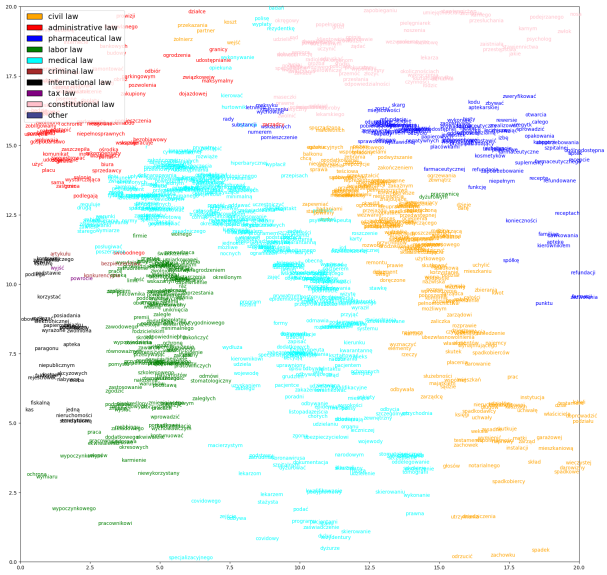

Figure 5. Strong keywords displayed on SOM

After these modifications, the structure of the space looked as illustrated in Figure 5. Table 3 shows that this treatment improved the coherence of the classes of documents examined.

4. Results and Experiments

4.1. Classification Accuracy

A classification accuracy of 71.69% was achieved during our experiments. This result, in terms of automatic classification of legal text, is satisfactory. A comparison of this outcome with random category selection, which resulted in only 13% accuracy, significantly underlines the advantage of the chosen method. It is worth noting that for less than 2% of the analyzed text, it was not possible to assign any category. This indicates the presence of segments that do not contain characteristic words for any of the legal categories.

4.2. Characteristics of the Dataset

The dataset used to train the model was significantly unbalanced. The categories "Constitutional Law" and "International Law", along with "Undefined", had fewer extracted words, which may potentially have reduced performance.

During testing, items related to these categories were removed, resulting in a slight increase in accuracy, reaching 72.03%.

It is worth noting that the obtained accuracy was significantly influenced by the characteristics of the dataset itself. The documents in this collection were unevenly distributed across different legal categories. For instance, "Medical Law" accounted for almost 30% of the total collection, while the share of "Constitutional Law" was only 0.15%. This uneven distribution of documents presents a potential challenge in the classification process, necessitating further analysis and optimization of the model.

4.3. Classification Based on Vector Comparison

This experiment was conducted to thoroughly investigate the effectiveness of an approach based on comparing the semantic similarity of words without using the NER model for entity recognition in the text. The main objective was to compare this method with traditional NER-based approaches and explore whether it can be equally, or even more effective in entity recognition.

The method consists of the following steps:

1) **Data Preparation:** Initial text processing, such as tokenization and stopword removal.
2) **Creation of Word Representation Vectors:** Words are encoded using GloVe, with the creation of numerical vectors for each word.
3) **Semantic Similarity Comparison:** Utilizes a similarity method to compare the similarity between words.
4) **Entity Classification:** For each word in the document, the word with the closest meaning is selected.
5) **Evaluation:** Similarity results for each category are averaged, and the category with the highest score is chosen.

The resulting accuracy obtained was approximately 46%, significantly lower than our original approach using NER, which was was around 72%. Here, usage of the domain–oriented tool for NER may improve the precision of the results achieved by the general purpose one.

4.4. Classification Based on the Vector Representation From the RoBERTa Transformer

The Robustly Optimized BERT Approach (RoBERTa) architecture model was used, pre-trained on a Polish corpus exceeding 200GB in size. The first step involved passing the name of each legal category through the model; the obtained embeddings had a length of 1024. Then, for each query, the title of the document was extended with the text "query:". The entire sentence, without any filtering, was then used as input to the model to obtain an embedding. Using cosine similarity, it was possible to determine which category the "query" best matched. The overall result achieved for all categories, shown in Table 4, was 56.8%.

Table 4. Comparison of accuracy between the NER model and Polish RoBERTa in each category of document

Category	RoBERTa [%]	NER [%]
Civil law	68.88	72.00
Administrative law	31.40	55.73
Pharmaceutical law	67.80	74.36
Labor law	62.88	59.15
Medical law	65.56	77.05
Criminal law	55.77	62.22
International law	0.00	35.29
Tax law	12.77	77.27
Constitutional law	13.33	50.00

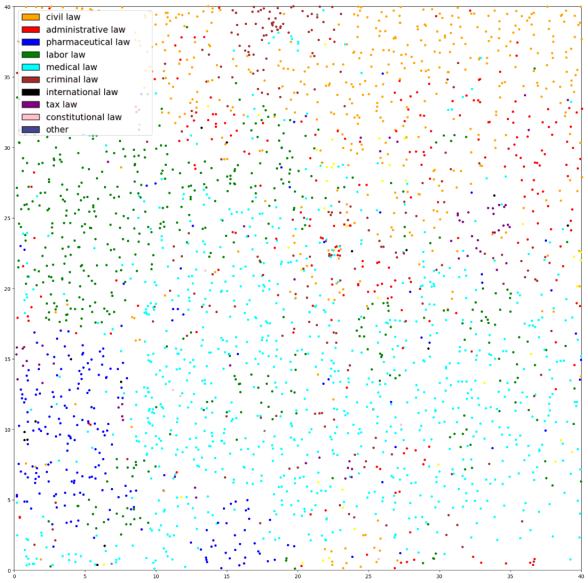

Figure 6. SOM obtained from RoBERTa embeddings

Figure 7. Strongest keyword per query from RoBERTa model

Table 5. Coherence (C) and descriptiveness (D) of classes using RoBERTa embeddings.

Category		C	D
Civil law	231	0.6	0.055
Administrative law	151	0.53	0.08
Pharmaceutical law	298	0.68	0.048
Labor law	583	0.6	0.023
Medical law	994	0.59	0.027
Criminal law	20	0.53	0.235
International law	137	0.61	0.076
Tax law	25	0.45	0.195
Constitutional law	280	0.64	0.056

Analyzing the accuracy for individual categories, only in the "labor law" category (59.15% vs. 62.88%) was a better result observed than with the NER model, which can be considered an anomaly since the improvement is not significant.

Since the vectors here are significantly longer than those used to generate the previous Kohonen networks, a 40×40 map was used in this case. The result is presented below in Figure 6. The classes on this diagram are much more separated and visible than in Figure 2, but the accuracy of the results was still inferior to NER. Extracting keywords from sentence embeddings isn't a perfect process, but by performing leave-one-out analysis and seeing which word, when missing, impacted the relative cosine similarity of the selected class compared to the next best, the strongest keyword for each query was extracted and is visualised below in Figure 7. The descriptiveness and coherence of the classes are summarized for each category in Table 5. While we still consistently saw worse results than NER, considering that zero preprocessing, filtering, and additional training were performed, they were very impressive.

5. Conclusion and Future Works

This paper aimed to develop an effective classification system for legal documents through keyword frequency analysis. The approach utilized the Kohonen self-organizing map method and word vector representation using the GloVe model, achieving a satisfactory classification accuracy of 71.69%.

The research conducted revealed that the classification process can be enhanced through the application of various techniques, such as NER tools and the RoBERTa model, to achieve similar accuracy without any domain-specific fine-tuning.

Challenges encountered during the process, such as the uneven distribution of categories in the dataset, highlighted areas for further improvement. Thus, future research should focus on experimenting with AI-based methods that handle unbalanced data more effectively, and exploring techniques such as generating artificial data or resampling to balance the dataset.

Notes

[1]https://nlp.stanford.edu/projects/glove/
[2]https://spacy.io/api/entityrecognizer

AUTHORS

Paulina Puchalska – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: paulina.puchalska@pg.edu.pl.

Kacper Krzemiński – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: kacper.krzeminski@pg.edu.pl.

Maksymilian Lis – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: maksymilian.lis@pg.edu.pl.

Rafał Scherer* – Częstochowa University of Technology, Generała Jana Henryka Dąbrowskiego 69, 42-201 Częstochowa, Poland, e-mail: rafal.scherer@pcz.pl, www: https://kisi.pcz.pl/rscherer.

Paweł Drozda – University of Warmia and Mazury in Olsztyn, Michała Oczapowskiego 2, 10-718 Olsztyn, Poland, e-mail: pdrozda@matman.uwm.edu.pl, www: wmii.uwm.edu.pl/kadra/drozda-pawel.

Kajetan Komar-Komarowski – Lex Secure, Niepodległości 723/6, 81-853 Sopot, Poland, e-mail: kkk@lexsecure.com, https://lexsecure.pl/.

Konrad Szałapak – Lex Secure, Niepodległości 723/6, 81-853 Sopot, Poland, e-mail: ks@lexsecure.com, https://lexsecure.pl/.

Andrzej Sobecki – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: andrzej.sobecki@pg.edu.pl, https://pg.edu.pl/p/andrzej-sobecki-64426.

Tomasz Zymkowski – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland, e-mail: tomasz.zymkowski@pg.edu.pl.

Julian Szymański – Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, e-mail: julian.szymanski@eti.pg.gda.pl, https://julian.eti.pg.edu.pl/.

*Corresponding author

ACKNOWLEDGEMENTS

This work was partially supported by funds allocated to the project of "A semi-autonomous system for generating legal advice and opinions based on automatic query analysis using the transformer-type deep neural network architecture with multitasking learning," POIR.01.01.01-00-1965/20.

References

[1] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi, "Simple Unsupervised Keyphrase Extraction using Sentence Embeddings", *arXiv e-prints*, vol. 1, 2018, 1–9, doi: 10.48550/arXiv.1801.04470.

[2] T. Y. Christyawan and W. Firdaus Mahmudy, "Text Classification and Visualization on News Title Using Self Organizing Map", *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2018, doi: 10.1109/SIET.2018.8693189.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *CoRR*, 2018, doi: 10.48550/arXiv.1810.04805.

[4] R. Dzisevič and D. Šešok, "Text Classification using Different Feature Extraction Approaches", *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2019, doi: 10.1109/eStream.2019.8732167.

[5] M. R. Faisal, I. Budiman, F. Abadi, D. T. Nugrahadi, M. Haekal, and I. Sutedja, "Applying Features Based on Word Embedding Techniques to 1D CNN for Natural Disaster Messages Classification", *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, 2022, doi: 10.1109/IC2IE56416.2022.9970188.

[6] E. Frank and R. R. Bouckaert, "Naive Bayes for Text Classification with Unbalanced Classes". In: *Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10*, vol. 1, 2006, 503–510.

[7] S. X. Gao Zhengjie, Feng Ao and W. Xi, "Target-Dependent Sentiment Classification With BERT", *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2946594.

[8] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, "Word Cloud Explorer: Text Analytics Based on Word Clouds", *2014 47th Hawaii International Conference on System Sciences*, 2014, doi: 10.1109/HICSS.2014.231.

[9] R. Jing, "A Self-attention Based LSTM Network for Text Classification", *IOP Publishing*, 2019, doi: 10.1088/1742-6596/1207/1/012008.

[10] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some Effective Techniques for Naive Bayes Text Classification", *IEEE transactions on knowledge and data engineering*, vol. 18, no. 11, 2006, 1457–1466.

[11] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, L. Id, and Barnes, "Text Classification Algorithms: A Survey", *Information (Switzerland)*, 2019, doi: 10.3390/info10040150.

[12] P. S. Navada Arundhati, Ansari Aamir Nizam and S. Balwant, "Overview of use of decision tree algorithms in machine learning", *2011 IEEE Control and System Graduate Research Colloquium*, 2011, doi: 10.1109/ICSGRC.2011.5991826.

[13] M. Osowski, K. Lorenc, P. Drozda, R. Scherer, K. Szałapak, K. Komar-Komarowski, J. Szymański, and A. Sobecki, "Previous Opinions is All You Need—Legal Information Retrieval System". In: *International Conference on Computational Collective Intelligence*, 2023, 57–67.

[14] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents", *Text Mining: Applications and Theory*, 2010, doi: 10.1002/9780470689646.ch1.

[15] F. P. Shah and V. Patel, "A Review on Feature Selection and Feature Extraction for Text Classification", *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, doi: 10.1109/WiSPNET.2016.7566545.

[16] A. Sun, E.-P. Lim, and Y. Liu, "On Strategies for Imbalanced Text Classification Using SVM: A Comparative study", *Decision Support Systems*, vol. 48, no. 1, 2009, 191–201.

[17] A. Talun, P. Drozda, L. Bukowski, and R. Scherer, "FastText and XGBoost Content-Based Classification for Employment Web Scraping". In: *International Conference on Artificial Intelligence and Soft Computing*, 2020, 435–444.

[18] J. W. Xuelian Deng, Yuqing Li and J. Zhang, "Feature Selection for Text Classification: A Review", *Multimedia Tools and Applications*, 2019, doi: 10.1007/s11042-018-6083-5.

[19] L. Yang, "A Brief Introduction of the Text Classification Methods", *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2022, doi: 10.1109/EEBDA53927.2022.9744845.

[20] G. Yenduri, M. Ramalingam, G. ChemmalarSelvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. DeeptiRaj, R. H. Jhaveri, B. Prabadevi, W. Wang, A. V. Vasilakos, and T. R. Gadekallu, "Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions", *ArXiv*, 2023, doi: 10.48550/arXiv.2305.10435.

[21] P. L. Ying Chen and C. P. Teo, "Regularised Text Logistic Regression: Key Word Detection and Sentiment Classification for Online Reviews", *arXiv e-prints*, 2020, doi: 10.48550/arXiv.2009.04591.

[22] Y. Zhang, "Research on Text Classification Method Based on LSTM Neural Network Model", *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2021, doi: 10.1109/IPEC51340.2021.9421225.