Illinois cases linguistic analysis

Davide Carletti and Francesco Tomaselli

University of Milan

Abstract. We here present an unsupervised methodology, based on topic modelling and word embeddings, to study temporal trends and correlations in time of single and group of words in legal texts.

Keywords: Legal texts · Topic modelling · Word embeddings

1 Introduction

The goal of this project is to study words among the temporal axis in court decisions. More precisely, we aim to find information about relevance and frequency of single or multiple terms, and also to discover correlations between them.

The work is accessible at https://github.com/tomfran/legal-texts-information-retrieval and can be tested by using the developed webapp at https://illinois-cases-analysis-webapp-qka7d4ktba-ew.a.run.app/.

2 Research question and methodology

The goal of the project is to analyze, from a collection of court decisions, the relevance of terms in time while also gathering information about possible correlations between them. The reasons to perform such a study are various, firstly, one could find interesting and unexpected correlations between terms, also, studying them in time could reveal a change of context of some words.
The research question is focused in particular on three main areas of interest: narcotics, weapons and investigations. Thus, words related to this three main sets are the most interesting ones to analyze.

The *Mining the Harvard Caselaw Access Project* paper has been used as a starting point for both the analysis and the visualization part, which explores a different area of interest but with some common ideas. [1]

2.1 Initial idea

In the first place, the methodology aimed to divide the dataset into the three main areas, corresponding to the area of interests previously introduced. Having documents divided in areas, would have made possible to perform topic modelling and language modelling on the three areas, possibly considering the temporal factor in the process.

This approach would have allowed to directly confront the three sets also in a temporal way, showing differences in the language and topics between words regarding drugs, narcotics and investigations respectively.

Some scientific literature has been explored and tested, based on the idea of *Guided topic modelling* but it has been soon discovered that the dataset comprehended many other topics other than the three main areas of interest. [2] Another drawback was the fact that even when finding promising results with the guided method, the three topics were somehow intersecting too much, probably because the area of interests are closely related to each other in real life.

2.2 Refined methodology

After coming to a dead end with the approach presented in the previous Subsection, we opted for a different overall methodology, shifting from guided topic modelling towards a classical one and performing the temporal analysis with an heavy use of word embeddings.

3 Experimental results

3.1 Dataset preprocessing

The dataset in use is the Illinois Bulk Dataset, that contains 183146 cases with 194366 judges opinions.

The first step of the preprocessing is to merge the opinions about a case into one, obtaining a single document for each dataset entry. Then, each document goes through a text cleaning and tokenization phase, where the first part is done with the help of regular expressions, while the second uses Spacy to obtain a list of terms in their lemmatized form.

3.2 Word of interest expansion

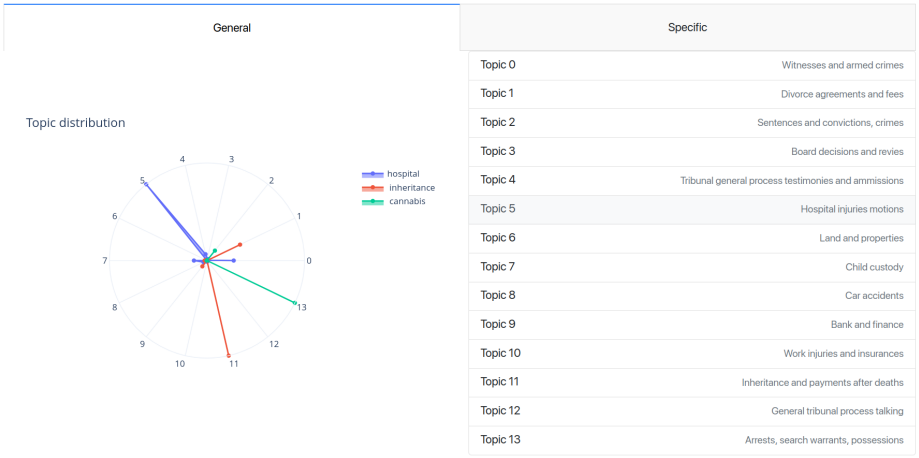
The project started with a collection of relevant words for three categories, namely weapons, narcotics and investigations. The idea is to preserve these words in each preprocessing phase, especially when filtering out words that do not meet a required frequency in the dataset. To have a better set of interesting words to keep we opted to expand them using pre-trained word embeddings, the GoogleNews models. The process consists of finding similar words for each word of interest, checking that these words are in the dataset, with a manual final review to remove unnecessary or wrong words.

3.3 Topic modelling

To have an overview of the topics discussed on the dataset a Latent Dirichlet Allocation model is trained on the tokenized texts. One of the key parameters of such a model is the number of topics, and, given the fact that cases could potentially talk about anything, an Halving search is performed to find a good value.

The basic idea is to find the optimal number of topics in a given range, here we try to find the best one between ten and thirty. A Grid search would test all topic numbers, making the process really intensive and slow. Halving search

mitigates the problem, as it trains firstly on smaller datasets, select the best models, and retrain on bigger slices of data until a final model is found. This methodology can be ten times faster than Grid search. The selection criterium for the search is the Log Likelihood of each model. The search revealed that the optimal number of topics is 14. An overview of the found topics can be seen on Figure 1.



Specific	
Topic 0	Witnesses and armed crimes
Topic 1	Divorce agreements and fees
Topic 2	Sentences and convictions, crimes
Topic 3	Board decisions and revies
Topic 4	Tribunal general process testimonies and ammissions
Topic 5	Hospital injuries motions
Topic 6	Land and properties
Topic 7	Child custody
Topic 8	Car accidents
Topic 9	Bank and finance
Topic 10	Work injuries and insurances
Topic 11	Inheritance and payments after deaths
Topic 12	General tribunal process talking
Topic 13	Arrests, search warrants, possessions

Fig. 1. Generic topic distribution for the words hospital, inheritance and cannabis

Results are promising but the words of interest, namely a collection of narcotics, investigation and weapons terms, are merged together in a few topics.

To solve the issue we decide to run topic modelling on a subset of the previous topics, considering only the ones more related to the three main areas of interest, with the same technique as before. The result is similar, we find again 14 topics, but this time they are much more specific, an example of specific topic distribution can be seen on Figure 2 on the next page, with an in depth topic analysis on Figure 3.

As stated on Subsection 2.1 on page 1, the first idea was to guide the topic modelling process to find three main topics, but the applied methodology failed on numerous occasions to perform well, we believe that running two phases of topic modelling resulted in a similar result, while discovering much more about the dataset.

3.4 Temporal word embeddings

One of the objective of the project is to find correlations among words and one tool that can be used is word embeddings. The technique assigns a real vector of a given dimensions to each word in the document collection, creating a way to directly compare the context similarity between two terms.

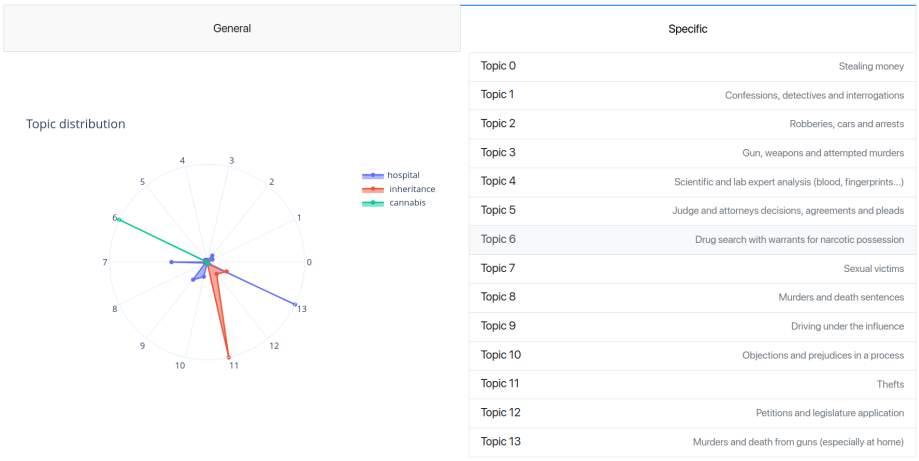


Fig. 2. Specific topic distribution for the words hospital, inheritance and cannabis

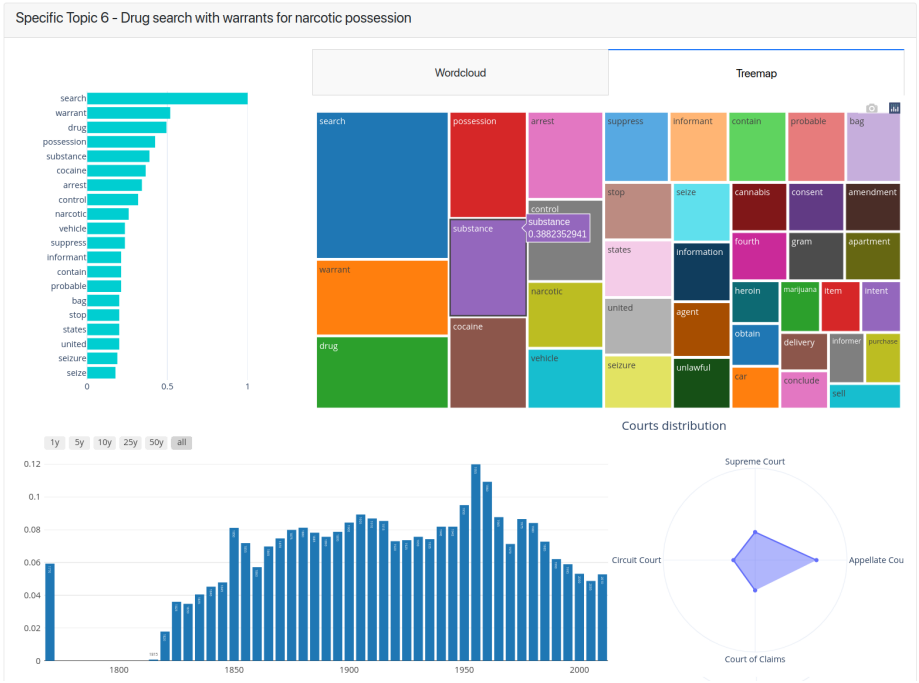


Fig. 3. Detailed topic analysis for the search warrant specific topic.

For this task we used Gensim's Word2Vec implementation and trained different models in three ways:

1. *Global model*: trained on the entire document collection, with 100 components vectors;
2. *One year models*: they are trained on subsets of the dataset divided by years, with 300 components;
3. *Ten years models*: trained on epochs of 10 years each, again with 300 components.

The first model gives information about the whole dataset, it can be used to compute similar words queries, while the others can be exploited to find context and semantic shifts among the temporal axis. Taking inspiration from the *HistWords* work on semantic shift, we start by aligning the models, and then find, given a word and a base year, the similarity of that word in time with respect to the base year. [3] Technically, we compute the cosine similarity of each year vector with respect to the vectors trained on the base year. This comparison is made possible by the initial alignment, it would not be meaningful otherwise. This approach can reveal if a word changed semantic or context, and when, with respect to a given year, an example can be seen in Figure 4.

Similarly, the ten year models are aligned, but this time, given a term, we compute the similarity among two consecutive epochs. The idea is similar to the previous one but slightly different, this time a drop in the similarity sequence reveals a change of meaning from an epoch to the other, the resulting sequence can be seen in Figure 5 on the following page.

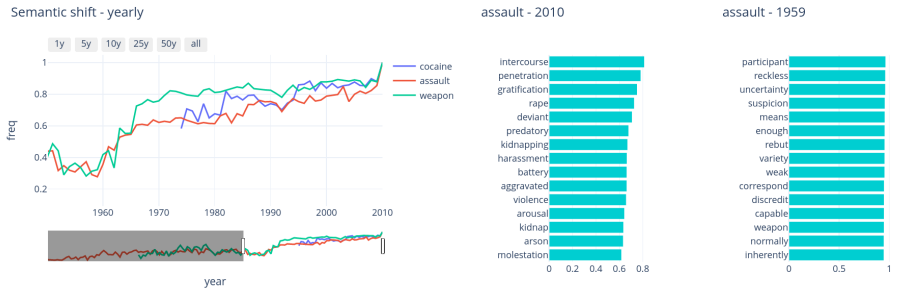


Fig. 4. Semantic shift of the word cocaine, assault and weapon with respect to the year 2010. A drop of the assault curve in 1959 correspond to a different context with respect to the base year, as proven by the different similar words.

3.5 Webapp

In order to visualize and explore the results of the analysis, a web interface has been developed. Through the UI, the user can search multiple words united,

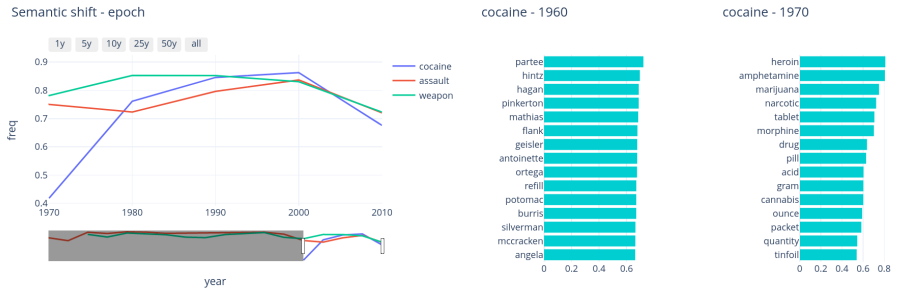


Fig. 5. Similarity between consecutive epochs of the terms cocaine, assault and weapon. Comparing the similar words, we can see that the drop of the cocaine curve around 1970 correspond to a shift in context.

separated by ",", and/or compare words, separated by "-"; the app will show various interesting sections, one focused on the analysis of the searched words:

- Top 15 words of similar context of the searched words, using the Global word embeddings model.
- Frequency of the single or group of searched words in time.
- Semantic shifts of the searched words, both by epoch and by single year. If the user clicks on a point of the graph, the similar words with respect to the query are shown, based on the year where the user clicked, making possible to evaluate the shift between epochs or years. Figure 5 shows the result of clicking year 1970.
- Topic distribution of the query, both generic and specific. Clicking a topic on the list or chart reveals an in-depth analysis in the subsequent section.
- Topic analysis section, showing most important words for a given topic, in a barchart, a Wordcloud and a Treemap. The year distribution and the relevance in the four different courts of the topic is also shown. Figure 3 on page 4 contains an example of the section.

3.5.1 Webapp developing technology

For the webapp development, three python-based frameworks for web developing have been considered, the first being the combination of Voila and ipywidgets. This would have been the natural candidates as the whole project has been developed using Jupyter notebooks. Ipywidgets allows to create dynamic notebook elements, while Voila can create a good design and permits to hide code cells. They have been discarded because of the technical requirements needed to run and understand a Jupyter notebook; even if the complexity could have been partially hidden, the graphics component of Ipywidgets are not so captivating.
The second options are Dash and Streamlit, they can create a point-&-click dashboard to visualize models, and they are the main solutions for data scientists and engineers in order to put complex Python analytics in the hands of business decision-makers and operators. While similar in terms of interface design,

Dash has been choose because of its popularity and presence in the data science applications world.

In all three frameworks, and particularly in Dash, the workflow is pretty similar: the design elements of the UI are defined in an HTML fashion using the framework Python libraries predefined classes. The user-generated events are captured using callback functions, which connect the input elements, e.g. a button which has been clicked, to output elements, e.g. a plot, defining the required processing and manipulation of output elements.

3.5.2 Cloud hosting

Since the developed webapp prototype shows some potential in finding interesting trends about words, we decided to host the project on a public domain, in order to give anyone the possibility to make queries without a technical knowledge or any installation requirement.

Firstly, many free-to-use hosting sites of Python web applications have been tested, such as PythonAnywhere and Heroku, but all of them allowed a to small free tier, for instance of one gigabyte of disk. Because of the large data that is used to show the various statistics, having models taking up to five gigabytes of space, it was not possible to use any of them.

Google Cloud Platform has been chosen to host the webapp. A Docker image of the project has been created, which has been built to Google Cloud Run, creating an online container of 16GB of memory and 4 CPUs, the last and only combination allowed which permitted also to load all the required files. Google Cloud offers a huge Free Tier, but because of the demanding configuration, the monthly cost is expected to be around three dollars. One downside of the selected hosting method is that the container instance is active only during the period in which requests are served, thus, if no other instance is active, the container has to be launched from an idle state, taking about three or four minutes to be accessible, due to the amount of files that it has to load in order to make all the analysis functionality available.

3.6 Interesting findings

We now present a collection of interesting findings with the developed tools. The webapp allows to search for single or combinations of words, and the data on context change and topics can reveal really interesting trends in time.

The first one is about murders, in particular the difference of context between murders involving men and women. In the first case, the topics associated with the combination murder-man are theft related, while in the second case, murder-woman is associated with the sexual victims topic. Searching for murder and rape reveals similarity with kidnaps and robberies.

A second finding is about the context change of the word drug, in fact, the ten year epoch similarity drops between 1940 and 1950, in the first case the word is associated with groceries, while in the second the word shifted towards a narcotic context.

If we associate the words drug and death we find that overdose is the closets word in term of vector distance.

Another example is the change of context of the word homosexual, in the 60s it is associated with words such as deviate, aggressive, unnatural, while in modern times it refers to sexual orientation.

Analyzing the word assault, we can see that in the 60s it was associated with armed crimes, while in modern days it has a sexual crime context.

In the most similar words with respect to home and murder combined we can find boyfriend, girlfriend and fiance.

Regarding courts and topics, we can

4 Concluding remarks

As showed in the previous subsection, it is possible to explore in various ways the dataset by exploiting a number of techniques, such as topic modelling, word embeddings, semantic shifts and words frequencies.

While some findings have been reported, and clearly it is possible and easy to think and search about other ideas, it is difficult to evaluate in a theoretical way the results of the analysis. In the literature, about legal text analysis, almost all methods consists of classification which rely on supervised methods, especially related to *querying systems*, exploiting the manual help of domain experts in building the dataset and evaluating the results. [4]

We decided to focus on an unsupervised methodology, trying to explore in a broad manner the dataset in order to extract as much latent information as possible. This work could serve as a starting point to expand what has been discovered, by taking advantage of the already implemented techniques such as topic modelling but considering new features of interest, or by implementing supervised techniques.

Some hints of possible future works could be the exploitation of the geographical areas of the judgments, the *type of opinions*, such as dissenting and consenting or even the individual judges to specialize the work done with topic modelling and word embeddings, possibly implementing new supervised techniques which could help to predict court judgments based on certain words and contexts. [5]

References

1. Felix B. Chang, Erin McCabe and James Lee. ACL 2020. Mining the Harvard Caselaw Access Project.
2. Jagarlamudi Jagadeesh, Daumé III Hal and Udupa Raghavendra. ACL 2012. Incorporating Lexical Priors into Topic Models.
3. William L. Hamilton, Jure Leskovec, and Dan Jurafsky. ACL 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change.
4. Jaromir Savelka, Huihui Xu and Kevin D. Ashley. ACL 2019. Improving Sentence Retrieval from Case Law for Statutory Interpretation.
5. Raquel Mochales and Marie-Francine Moens. ACL 2011. Argumentation Mining.