

# An Anonymization Tool for Open Data Publication of Legal Documents

Arttu Oksanen<sup>1</sup>, Eero Hyvönen<sup>1,2</sup>, Minna Tamper<sup>1,2</sup>, Jouni Tuominen<sup>1,2,3</sup>,  
Henna Ylimaa<sup>4</sup>, Katja Löytynoja<sup>4</sup>, Matti Kokkonen<sup>4</sup> and Aki Hietanen<sup>5</sup>

<sup>1</sup>Aalto University, Dept. of Computer Science

<sup>2</sup>University of Helsinki, Helsinki Centre for Digital Humanities (HELDIG)

<sup>3</sup>University of Helsinki, Helsinki Institute for Social Sciences and Humanities (HSSH)

<sup>4</sup>Statistics Finland

<sup>5</sup>Ministry of Justice in Finland

## Abstract

The EU General Data Protection Regulation (GDPR) requires anonymization of documents containing personal data, such as court decisions, for public use. Doing this manually is costly and time-consuming but can be automated by applying Natural Language Processing (NLP) methods. This paper introduces the ANOPPI tool developed for (semi-)automatic anonymization of Finnish texts. The tool can be used both as a web application and programmatically through a REST API. Evaluation shows that ANOPPI performs well with different types of documents, however, further improving the performance of the named entity recognition and disambiguation methods would enhance the usefulness of the software. The tool is being published as open source for public use by the Ministry of Justice in Finland. A use case of ANOPPI is to publish court decisions on the Web in the LawSampo semantic portal for human close reading and as Linked Open Data for data analysis in legal informatics.

## Keywords

anonymization, pseudonymization, case law, named entity recognition

## 1. Introduction

Many texts and legal documents of interest to a wider audience, such as interviews of people and court decisions, contain sensitive personal data. This makes it difficult to publish and use them given the EU General Data Protection Regulation (GDPR)<sup>1</sup>, unless personal data contained is disguised, which is costly and time-consuming to do manually.

This paper presents the software architecture and first evaluation results of the ANOPPI tool and web service<sup>2</sup> for automatic and semi-automatic anonymization of Finnish documents, extending our earlier short paper [1] about the tool. Utilizing both machine learning (ML) and

---

*Joint Proceedings of ISWC2022 Workshops: the International Workshop on Artificial Intelligence Technologies for Legal Documents (AI4LEGAL) and the International Workshop on Knowledge Graph Summarization (KGSUM) (2022)*

✉ arttu.oksanen@aalto.fi (A. Oksanen); eero.hyvonen@aalto.fi (E. Hyvönen); minna.tamper@aalto.fi (M. Tamper); jouni.tuominen@helsinki.fi (J. Tuominen); katja.loytynoja@stat.fi (K. Löytynoja); matti.kokkonen@stat.fi (M. Kokkonen); aki.hietanen@om.fi (A. Hietanen)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup><https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/>

<sup>2</sup>Project homepage: <https://seco.cs.aalto.fi/projects/anoppi/>

rule-based named entity recognition and linking methods (NER/NEL) [2] and morphological analysis, ANOPPI is able to automatically or semi-automatically anonymize documents written in Finnish while preserving their readability and layout.

ANOPPI is the first anonymization tool developed for Finnish. The tool was developed for automatic anonymization of court decisions [1] in the Anoppi project funded by the Ministry of Justice in Finland but can and has been used in other contexts, too. The source code will be published with an open license after the ongoing deployment process of the service is finished.

This paper first introduces the ANOPPI tool and the underlying method for automatic anonymization from an end-user point of view (Section 2). We then present the underlying technical ideas of the tool starting with a description of the software architecture, language analysis component, and the workflow in Section 3. After this, first results of evaluating the performance of ANOPPI in fully automatic anonymization are presented. In conclusion, related works and future directions of development are discussed.

## 2. ANOPPI Client User Interface

The screenshot shows the ANOPPI web interface. The top navigation bar includes the ANOPPI logo and a status message: "Tämä sovellus on vain testikäyttöön, älä käytä muita kuin Anoppi-hankkeen laatimia testitilaisiä!". Below the navigation bar, the interface is split into two main areas. The left area displays a document snippet with highlighted entities, including names like "Mynämäen seutukunta" and "Mietolisen kaupunki". The right area contains a control panel with a table of entities and a list of organizations. The table of entities has columns for ID, Perusmuoto, Korvaava teksti, and other details. The list of organizations has columns for ID, Perusmuoto, Korvaava teksti, and other details.

Figure 1: ANOPPI client user interface

The web-based user interface (WUI) of ANOPPI is depicted in Fig. 1. It is an HTML5/JavaScript application implemented with React user interface library. NPM (Node package manager) modules redux and redux-observable are being used for handling the application state and asynchronous logic.

The general idea of the tool is to find automatically named entities in the document and mark them in different colors and using different symbols based on their type (person, place, organization, etc.). A numeric ID is used to identify mark and disambiguate entities; multiple

occurrences of the same entity have the same ID. The marked document is shown on the left in Fig. 1. After finding the named entities and their occurrences in the text, the occurrences are replaced with referent identifiers. To assign a reasonable identifier for a given named entity its category must also be resolved. For example, we must be able to differentiate towns and corporations so that an identifier can be correctly determined as either “town A” or “corporation A”. Categorical disambiguation is based on a scoring scheme that weighs the results obtained from the different named entity recognizers.

On the right hand side of Fig. 1, each entity found is listed in a table that can be used for editing the entities and for specifying how the entity label is replaced in the anonymized document. For example, wrongly identified entities can be deleted and missing new ones be inserted, their type can be changed, the replacement string can be edited, and so on. By clicking on a “preview” button the final result can be seen and finally saved with its original formatting.

The court orders to be anonymized are available in electronic format either as plain text, XML, HTML, or DOCX files. Based on Natural Language Processing (NLP) tooling discussed in [3], we have developed a tool that is able to find the named entities from these documents and annotate the occurrences of the named entities with special tags.

### 3. ANOPPI Architecture

ANOPPI is a web-based tool for automatic or semi-automatic anonymization. It comes in two versions:

1. **Web application version** that has a user interface and the anonymization is done with manual checks and editing performed by a human user. The interface was described in the previous section.
2. **Standalone REST API version** that can be used programmatically for automatic or semi-automatic anonymization.

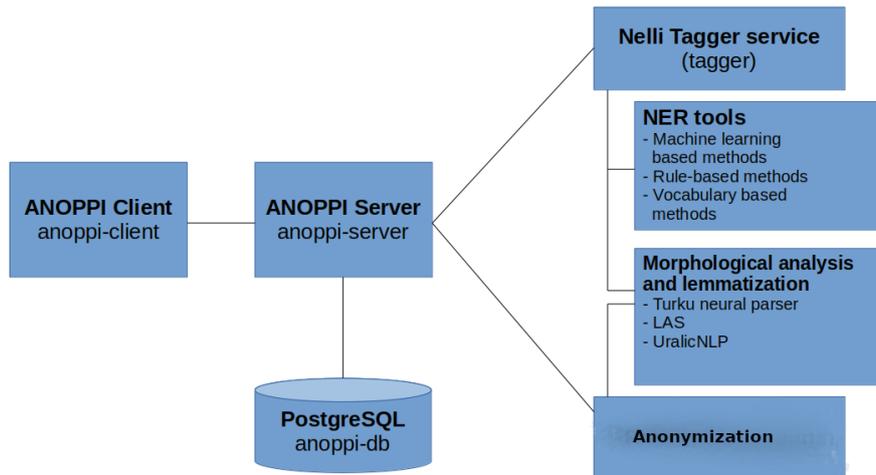
#### 3.1. Software Architecture

The Web application version depicted in Fig. 2 consists of a WUI (ANOPPI Client on the left), a back-end ANOPPI Server. The ANOPPI Server consists of language analysis components (LAC) that perform named entity recognition (NER) and anonymization of the text. The NER process is carried out by the Nelli Tagger Service [3, 4] that performs the morphological analysis, NER, and disambiguation on the text using different types of NER tools that apply rule-based methods, machine learning based methods, and vocabulary based methods. The results are then displayed in the ANOPPI client where the user can modify the result of the annotation. After modifications the document is anonymized by ANOPPI Server. All of the individual components are packaged as Docker containers. The whole system can run on a single machine and a single closed network.

The Web application version of the ANOPPI Server is implemented using Scala and Play web framework<sup>3</sup>. The back-end uses the Nelli Tagger Service to find the mentions of named

---

<sup>3</sup><https://www.playframework.com>



**Figure 2:** Components of the ANOPPI Web application

entities in text when a new document is uploaded to ANOPPI. In addition, LAS (SeCo Lexical Analysis Services) [5] and Uralic NLP [6] are used to morphologically analyze and lemmatize a piece of text when eg. creating a new named entity or generating the correctly inflected form of a referent identifier. The back-end stores the uploaded documents in a relational database (PostgreSQL<sup>4</sup>). The service generates UUID identifiers for the uploaded documents. These identifiers are stored in the database as well as the local memory of the user’s web browser. Currently the uploaded documents are connected to the user’s session and there is no separate user authentication mechanism. After the browser cache is emptied the uploaded documents can no longer be accessed.

The Standalone REST API version is similar to Fig. 2 but does not include a user interface client nor stores any user data in a database. The API can be used programmatically for either automatic or semi-automatic anonymization. It takes as input the document and produces as output the annotated or anonymized document with a separate list of all the named entities found in the document. A public test instance of this API<sup>5</sup> runs on the CSC Rahti container cloud. API documentation, based on Swagger, is available on the Web<sup>6</sup>.

The public API includes, for example, the following endpoints consumed by the ANOPPI Client:

**POST /project** Create a new project (document to be anonymized). The endpoint is consumed when uploading a new document to the web application. This function sends the document content to the tagger that finds mentions of named entities in the text. This tagged text is then returned back to the client.

**POST /project/preview** This endpoint is consumed by the client when previewing the

<sup>4</sup><https://www.postgresql.org/>

<sup>5</sup>The API can be found at <https://nlp.ldf.fi/anoppi>.

<sup>6</sup>Documentation: [https://app.swaggerhub.com/apis-docs/apoksane/open-api\\_nlp\\_ldf\\_fi/1.0.0](https://app.swaggerhub.com/apis-docs/apoksane/open-api_nlp_ldf_fi/1.0.0).

anonymized version of the document. It replaces the selected named entities in the text with their referent identifiers.

**POST /project/export** This endpoint is consumed by the client when exporting the finished anonymized version of the document. This endpoint prepares the final anonymized version of the document and sends it back to the client.

**POST /project/tag-entity-occurrences** Find mentions of a specific entity in text. This endpoint is used when user manually selects a new named entity from the text to be anonymized.

**GET /text/analyze** Analyze text. This endpoint is invoked after selecting a new name (a surface form) from the document text. The function resolves the lemma and case of the surface form. The required morphological analysis is done by utilizing LAS.

## 3.2. Language Analysis Components

In order to produce the automatically anonymized document, the ANOPPI server calls first for the Nelli Tagger Service to perform the NER process to find the named entities from the text. After obtaining the NER results, the anonymized version of the document can be produced using the anonymization component. The Nelli Tagger and anonymization process are presented in more detail in the following subsections.

### 3.2.1. Nelli Tagger Service

The identification of named entities in Finnish language documents can be a challenging task. The documents can contain not only regular names but also unusual names and registry numbers that are often not picked up by rule or machine learning based NER tools. For example, for both types of NER tools it can be difficult to extract foreign person names when processing documents that are, for example, related to immigration or refugees. Similarly, typically the ready-made tools do not pick registry numbers such as car plates or insurance numbers.

To find the named entities the Nelli Tagger Service [4, 3] is used. It uses an ensemble method that combines results obtained from multiple different named entity recognizers. First of all we use ready-made statistics- and rule-based named entity recognition (NER) software, such as FinBERT's NER model<sup>7</sup> [7], a rule-based named entity recognizer for Finnish language, and Stanford NER [8]. Secondly, we have developed our own set of regular expression patterns to recognize things such as vehicle registration plates and property identifiers [9]. In addition, we use an all-inclusive Finnish person name ontology [9] that is based on the open data published by the Population Register Centre<sup>8</sup> to look up person names appearing in the court cases. Finally, we use the Finnish Turku Neural parser [10, 11] to support deciding if a term appearing in the text is a name. The FinBERT NER model and Finnish Turku Neural parser were selected for their performance. The other NER tools complement the results of the FinBERT NER model's results by adding entities such as rare names that cannot be identified with the pretrained models.

After performing NER, the results are disambiguated by scoring results of each NER method's interpretation of the text, where the most popular interpretation wins. The results are then converted to JSON format that consists of a list of entities and their features, the original text

---

<sup>7</sup><https://turkunlp.org/fin-ner.html>

<sup>8</sup><https://vrk.fi/en/>

annotated with named entities using HTML span tags, as well as metadata about the run of Nelli Tagger Service (e.g., timestamp, success or error codes). The Nelli Tagger Service’s API is available as a REST API<sup>9</sup>. This service has been previously evaluated to perform with 86 % accuracy [3], however, since the evaluation, tools have been updated as the FinBERT NER model has been added that can perform NER with 93 % accuracy [4, 7].

### 3.2.2. Anonymization

The anonymization is done by replacing the recognized names with grouped sequential identifiers, such as ‘A’ and ‘B’ for persons or ‘place A’ and ‘place B’ for places, keeping track of the entities throughout the text. Using capital letters as referent identifiers was chosen to mimic the way anonymization is currently carried out manually.

As Finnish is a highly inflected language the correct inflected form for the referent identifier has to be derived to enhance readability of the anonymized version of the text. For this reason, morphological analysis is being used to distinguish, for example, the case and possessive suffix of nouns. Turku Neural Parser [11] is used to perform morphological analysis of the original text and UralicNLP<sup>10</sup> [12] to correctly inflect the referent identifiers to match the original text. Inflection is done using a finite state transducer (FST) model.

### 3.3. Workflow

The flow diagram in Fig. 3 illustrates how the data is processed in the ANOPPI system during different steps of the anonymization process.

As presented in Fig. 3 the anonymization process starts from the left when the user uploads a new document to ANOPPI. Firstly, the document is loaded to the ANOPPI Server and the original document is stored in the database. Then the text is extracted from the document and sent to the Nelli Tagger Service. The service performs named entity recognition and named entity disambiguation (NED) on a document. Here, the Nelli Tagger Service aims to maximize recall in the named entity recognition phase because it is easier for the user of WUI to delete suggested entities than to manually pick new entities from the text, if LAC did not automatically recognize them. It is also possible to restrict anonymization to certain types of entities (for example only person names). Further fine-tuning can be done by whitelisting certain names and contexts, for example to keep names of the judges overt in court judgments.

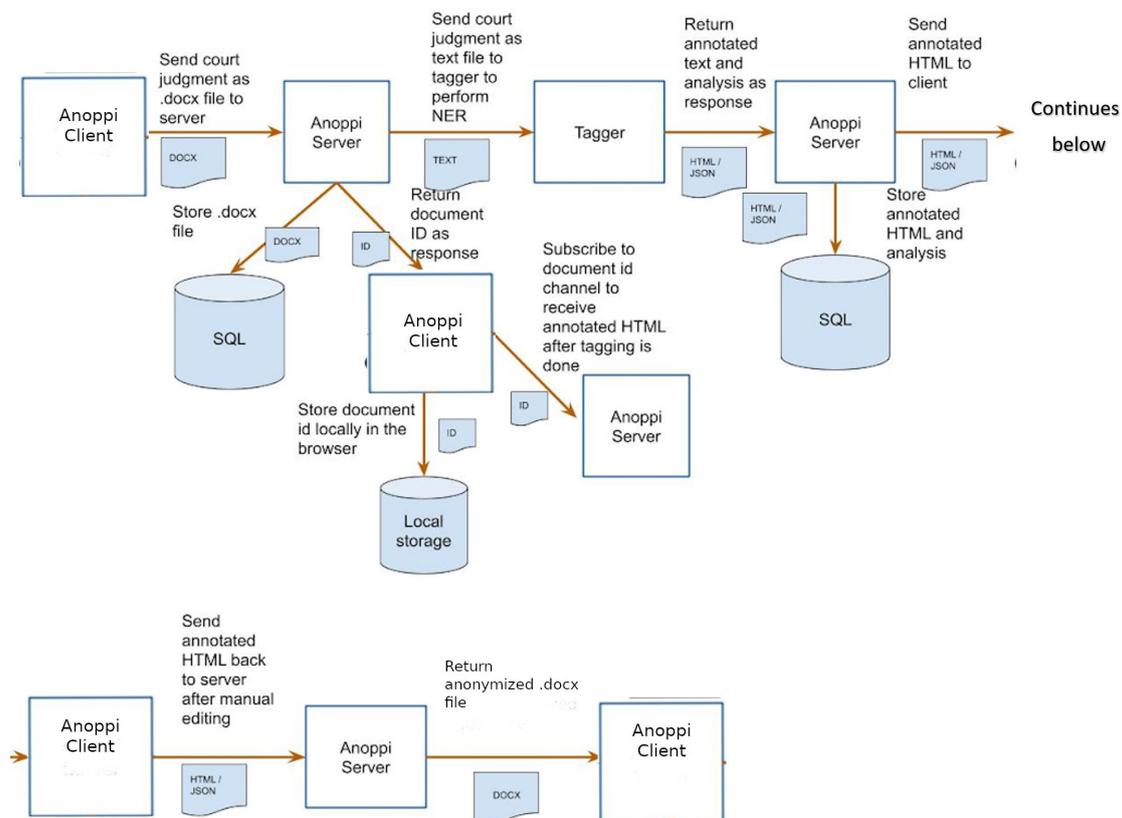
Once the Tagger Service returns the entities to the server the annotated result is recorded into a database for later usage and sent to the client in HTML format. The user checks the results in the WUI and can correct and preview the result. Once the client is satisfied with the result, the client can export the anonymized document. As the original DOCX file was stored initially the anonymized document can be created by fetching the original file from the database and switching its content with the anonymized one, preserving the original layout of the file.

The database holds the records of the document and in case the same document is requested again, the document can be dug up from the database instead of reprocessing it again with the

---

<sup>9</sup><https://app.swaggerhub.com/apis-docs/SeCo/nlp.ldf.fi/1.0.0>

<sup>10</sup><https://uralicnlp.com/>



**Figure 3: ANOPPI Workflow overview**

LAC tagger component. However, old documents are scheduled to be deleted from the database after a configurable period of time.

## 4. Evaluation

To test the performance of the LACs in anonymization of persons we created a tailor-made test dataset that consists of text in Finnish with person names added in grammatically appropriate places. The names were handpicked from the Population Information System data with emphasis on selecting both common and rare names. Attention was paid to select names including both traditional Finnish names and foreign names as well as names with two parts and names with a common meaning, for example Karhu (bear). Investigation reports from Safety Investigation Authority of Finland (SIAF) were used as base text for the test data. These reports do not originally contain any names but only references to people in the form of pronouns and job titles that were replaced randomly by the selected names and their combinations.

Eventually the test data contained 152 added names and name combinations from which

ANOPPI identified and anonymized 136 (89,5 %) correctly. In total, ANOPPI identified 141 names or name combinations from the test data, but five of them were false positive words. A total of 16 names remained unidentified. However, most of the unidentified names were located in parts of the text where names are not common which would not be an issue with real data.

In addition to evaluating the performance of the LAC, the ANOPPI tool as a whole has been evaluated by measuring and comparing the time it takes to anonymize a court decision both semi-automatically using the WUI and manually using only word processor software. The results obtained in this manner so far show that on average it takes about half the time to anonymize a court order using ANOPPI as compared to manual anonymization. ANOPPI makes some mistakes especially by confusing person names with place names and vice versa and spotting and correcting all of these incorrect categories using the WUI slows down the process. Moreover, in order to verify the correctness of the anonymization result a human expert still has to get acquainted with the content of the document regardless of the anonymization being automatic.

## 5. Related Works and Discussion

Automatic or semi-automatic anonymization and pseudonymization<sup>11</sup> methods are already in use in several European judicial systems [13]. For example, in Denmark an anonymization tool for court orders was implemented using solely manually crafted grammar rules to find the named entities in the texts [14]. Recent projects similar to ours focusing on automatic anonymization of court orders using ML methods possibly in combination with rule-based ones have been conducted for example in Poland, Austria, Germany, Latvia and France<sup>12</sup>. Regarding the NER/NEL services embedded in ANOPPI there are lots of related research surveyed in [2]; our approach and system with related works were discussed in Section 3.2.

Evaluation of ANOPPI shows promising results in locating the names of persons, organizations, places, and different types of identifiers of specific form. Still, it is difficult to build a general solution for anonymization as the sufficiency of de-identification varies in each case. The category-based selection of named entities used in the current model is not sufficient if for example names of small companies should be anonymized but large ones should not. Another issue in the ANOPPI project is the lack of task-specific training data as we are not able to store and make use of real production data in order to continuously train ML models due to restrictions imposed by the GDPR. That is why we ended up using a general NER model for Finnish language along with configurable case-based rules.

The ANOPPI service is currently in pilot testing in the Ministry of Justice of Finland for anonymization of Finnish court decisions in order to make them available on the Web and for data analysis in the forth-coming public LawSampo data service and portal for publishing and studying Finnish legislation and case law. Future work focuses on further improving the performance of the named entity recognition algorithm and including identification of new

---

<sup>11</sup>See <https://tietosuoja.fi/en/pseudonymised-and-anonymised-data> for the difference between the notions of anonymization and pseudonymization.

<sup>12</sup>Based on webinar presentations at: [https://ec.europa.eu/info/policies/justice-and-fundamental-rights/digitalisation-justice/conferences-and-events\\_en#webinarsontheuseofartificialintelligenceinthejusticefield](https://ec.europa.eu/info/policies/justice-and-fundamental-rights/digitalisation-justice/conferences-and-events_en#webinarsontheuseofartificialintelligenceinthejusticefield)

entity types such as rare diseases or unique job titles that make re-identification of people straightforward.

**Acknowledgments** This work is part of Finnish AI special funding program by the Ministry of Finance, for experiments that promote productivity. CSC – IT Center for Science, Finland provided computational resources.

## References

- [1] A. Oksanen, M. Tamper, J. Tuominen, A. Hietanen, E. Hyvönen, Anoppi: A pseudonymization service for Finnish court documents, in: *Legal Knowledge and Information Systems. JURIX 2019: The Thirty-second Annual Conference* (Araszkievicz, M. and Rodriguez-Doncel, V. (eds.)), IOS Press, 2019, pp. 251–254.
- [2] J. L. Martinez-Rodriguez, A. Hogan, I. Lopez-Arevalo, Information extraction meets the semantic web: A survey, *Semantic Web – Interoperability, Usability, Applicability* 11 (2020) 255–335. doi:10.3233/SW-180333.
- [3] M. Tamper, E. Hyvönen, P. Leskinen, Visualizing and analyzing networks of named entities in biographical dictionaries for digital humanities research, in: *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICling 2019)*, Springer-Verlag, 2019. Forthcoming.
- [4] M. Tamper, A. Oksanen, J. Tuominen, A. Hietanen, E. Hyvönen, Automatic annotation service appi: Named entity linking in legal domain, in: A. Harth, V. Presutti, R. Troncy, M. Acosta, A. Polleres, J. D. Fernández, J. Xavier Parreira, O. Hartig, K. Hose, M. Cochez (Eds.), *The Semantic Web: ESWC 2020 Satellite Events*, volume 12124 of *Lecture Notes in Computer Science*, Springer-Verlag, 2020, pp. 208–213. URL: [https://doi.org/10.1007/978-3-030-62327-2\\_36](https://doi.org/10.1007/978-3-030-62327-2_36). doi:10.1007/978-3-030-62327-2\_36.
- [5] E. Mäkelä, LAS: an integrated language analysis tool for multiple languages, *The Journal of Open Source Software* 1 (2016).
- [6] M. Hämäläinen, UralicNLP: An NLP library for Uralic languages, *Journal of Open Source Software* 4 (2019) 1345. doi:10.21105/joss.01345.
- [7] J. Luoma, M. Oinonen, M. Pyykönen, V. Laippala, S. Pyysalo, A broad-coverage corpus for Finnish named entity recognition, in: *Proceedings of the 12th Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, 2020, pp. 4615–4624. URL: <https://aclanthology.org/2020.lrec-1.567>.
- [8] J. Rose Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by Gibbs sampling, in: *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics*, Proceedings of the Conference, 2005, pp. 363–370.
- [9] M. Tamper, P. Leskinen, J. Tuominen, E. Hyvönen, Modeling and publishing finnish person names as a linked open data ontology, in: *3rd Workshop on Humanities in the Semantic Web (WHiSe 2020)*, CEUR Workshop Proceedings, vol. 2695, 2020, pp. 3–14. URL: <http://ceur-ws.org/Vol-2695/paper1.pdf>.
- [10] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä, S. Ojala, T. Salakoski, F. Ginter, Building the essential resources for Finnish: the Turku Depen-

- dency Treebank, *Language Resources and Evaluation* 48 (2014) 493–531. doi:10.1007/s10579-013-9244-1, open access.
- [11] J. Kanerva, F. Ginter, N. Miekka, A. Leino, T. Salakoski, Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task, in: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Association for Computational Linguistics, 2018.
- [12] M. Hämäläinen, UralicNLP: An NLP library for uralic languages, *Journal of Open Source Software* 4 (2019) 1345. doi:10.21105/joss.01345.
- [13] M. van Opijnen, G. Peruginelli, E. Kefali, M. Palmirani, On-Line Publication of Court Decisions in the EU: Report of the Policy Group of the Project 'Building on the European Case Law Identifier', 2017. Available at SSRN: <https://ssrn.com/abstract=3088495>.
- [14] C. Povlsen, B. Jongejan, D. H. Hansen, B. K. Simonsen, Anonymization of court orders, in: *11th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Las Palmas, Spain, 2016. doi:10.1109/CISTI.2016.7521611.