


Extracting Geographical References from Finnish Literature


Fully Automated Processing of Plain-Text Corpora

Harri Kiiskinen¹ 

Asko Nivala² 

Jasmine Westerlund² 

Juhana Saarelainen² 

1. Rebase Consulting, Finland.
2. Department of Cultural History, University of Turku , Turku, Finland.

Citation

Harri Kiiskinen, Asko Nivala, Jasmine Westerlund, and Juhana Saarelainen (2023). "Extracting Geographical References from Finnish Literature. Fully Automated Processing of Plain-Text Corpora". In: *Journal of Computational Literary Studies* 2 (1).

[10.48694/jcls.3584](https://doi.org/10.48694/jcls.3584)

Date published 2024-01-31

Date accepted 2023-11-23

Date received 2023-01-27

Keywords

named entity recognition, geographic information system, geoparsing, linked open data, literary geography, Finland

License

CC BY 4.0 

Reviewers

Roxana Patraş, Christof Schöch

Note

This paper has passed through the conference track of JCLS. In addition to being peer reviewed, it was presented and discussed at the 2nd Annual Conference of Computational Literary Studies at Würzburg University in June 2023.

Abstract. In the *Atlas of Finnish Literature 1870-1940* project, we extract geographical information from a Finnish-language corpus of literary texts published between 1870 and 1940. The texts are transformed from plain texts to TEI/XML, and further processed with named entity recognition and linking tools. The results are presented in a web-based environment. This article describes the technical structure of the analysis chain, the tools used and the metaprocesses used to manage the research dataset.

1. Introduction

1.1 Project

Literary geography asks where literature is located and why. The term was already in use in the early 1900s (for the definition and history of literary geography: Piatti 2008, 20, 65–121). As a modern research paradigm, literary geography emerged in the late 1990s as part of the spatial turn. Franco Moretti's *Atlas of the European Novel* can be considered a classic of this approach (Moretti 1998). Literary geography can also use cultural geography methods and qualitative analysis, in which case it does not necessarily make use of maps (e.g. Tally 2018). However, it often aims to map place names or other variables, in which case it can also be called literary cartography.

Place names and spatial information can be annotated and collected manually from texts. However, language technology allows spatial information to be extracted using computational methods. In the 2000s, literary geography has adopted the methods of digital humanities, combining natural language processing (NLP) with geographic information systems (GIS). Gregory and Hardie have used a part of speech tagger (POS) to extract proper nouns from the *Lancaster Newsbooks Corpus* (1653-1654) and filtered the results to place names using a gazetteer. They have then imported the place names onto maps using GIS and used density maps to analyse the results (Gregory and Hardie 2011). In a similar project, works describing the English Lake District were parsed for place names and the place names were references using a Gazetteer (Gregory et al. 2019). The Edinburgh Geoparser that was also used in these projects has been developed further since (Alex et al. 2019).

Named entities in literature have been studied mostly in English-language texts, but research has also extended to smaller language areas. For example, in the context of EL-TeC (European Literary Text Collection), the relevant COST Action has done significant work on the study of named entities in non-English European literature (Frontini et al. 2020). Transformers and BERT language models have significantly improved Named Entity Recognition (NER) tasks in non-English languages due to their ability to capture contextual information and learn representations from vast amounts of unlabeled text (Labusch et al. 2019). By employing self-attention mechanisms, transformers can effectively model long-range dependencies and capture intricate patterns in the data. BERT, in particular, introduced the concept of pretraining and fine-tuning, enabling the model to learn from massive amounts of text before being fine-tuned on specific NER tasks (Devlin et al. 2018). This approach has led to substantial advancements in NER performance also in Finnish language datasets (Luoma et al. 2020).

Our *Atlas of Finnish Literature 1870-1940* project applies similar methods for the first time to the study of Finnish literature. The objectives of the project are:

- to recognise place names mentioned in Finnish literature from the 1870s to the 1940s;
- to store geo-annotated texts and their metadata in a database;
- to geocode and enrich the identified place names by linking them to linked open data (Wikidata, DBpedia, Open Street Map etc.);
- to publish interactive maps showing the locations mentioned.

NER can be used to extract proper nouns – such as people, places and organisations – from unstructured text. As our project focuses on literary geography, we are mainly interested in political and geographical toponyms, as well as street and building names. After the recognition process, the locations are disambiguated, and geographic coordinates are retrieved by linking them to the linked open data. The location names can then be enriched using semantic web databases such as DBpedia and Wikidata. This allows, for example, the separation of cities from natural landscapes (e.g. rivers, lakes and mountains).

The aim of our project is to present a new interpretation of the geography imagined by Finnish-language literature in 1870–1945. The identification of named spatial entities offers a new method of exploring the territories to which fictional texts refer that has not been previously applied in Finnish scholarship. Based on the preliminary results, it seems that during the period under study, spatial references in literature were particularly related to nation-building and the definition of the territory of Finnish culture in relation to Sweden in the west, Russia in the east and the Sámi regions in the north. Historical novels seem to contain considerably more references to existing place names than other genres. Germany, Italy and France were the settings for many of the novels, and the migration from Finland to America is also reflected in the literature, but there are surprisingly few references to the British Isles. On the other hand, the polyphonic nature of literature should be emphasised: For example, labour literature had different aims from nationalist historical novels. However, we will present the literary-historical results in future publications once the mapping and historical analysis of the data has

been completed. The purpose of this paper is to describe the construction of the computational infrastructure of the study: corpus design, NER, NEL and the production of annotated XML-TEI.

In this paper, we will describe the text corpus used and the methods applied and developed in the project. In the remainder of the introduction, we describe the characteristics of Finnish literary history, our two corpora and the principles used to collect and verify the metadata. In [section 2](#), we then describe the automated process by which the plain text is converted into XML-TEI format, segmented into discrete works and chapters, and how the named entities are identified and linked. In [section 3](#), we describe how the data processing has been implemented with the message-broker system and what kind of manual checks have been made on the results to control the accuracy of the results.

1.2 Literary Historical Context and Primary Sources

Finland is a bilingual country that was part of Sweden from the Middle Ages to 1809. After the Finnish war 1808–1809, Finland was annexed to Russia and became an autonomous Grand Duchy. Finnish was the language of common people, even though the first printed books in Finnish were published already in the 1540s. Until the end of the nineteenth century, the great majority of the Finnish intelligentsia were still Swedish-speakers. The century was a time of modernisation for the Finnish language. Both Finnish and Swedish were defined as national languages in 1919.

At this stage, our project only covers texts written originally in Finnish. As said, Finland is a bilingual country, and Swedish-language fiction was also published during the period 1870-1940. In the future work stages of our project, we intend to study them as well. However, named entity recognition in Swedish requires the development of a separate NER and disambiguation system. For this reason, we limit our research at this stage to Finnish-language fictional works – including novels, dramas and poems. A comparison of Finnish- and Swedish-language literature could reveal interesting results concerning, for example, whether Finnish literature in Swedish describes more southern cities, coastal areas and archipelagos. On the other hand, Swedish-language material would also provide an opportunity to study a much earlier historical period when no significant Finnish-language literature was published. We will endeavour to carry out this study at a later stage.

Seitsemän veljestä (*Seven Brothers*, 1870) by Aleksis Kivi is usually considered as the first novel written in Finnish. The year 1870 is also the starting point of our research project. Additionally, it has been also suggested that it was only by circa 1880 that Finnish had developed into its modern form as a literary language, previous decades 1810-1880 being labelled as “era of Early Modern Finnish” and the time before that as “old literary Finnish”. Therefore, our project focuses on first decades of Finnish taking its modern shape. Finnish literature underwent an enormous increase and progress during the period we study. Many new writers, including Minna Canth, Juhani Aho, Arvid Järnefelt and Santeri Alkio, rose to popularity. The Finnish nationalism and the Fennoman movement actively promoted Finnish language and literature. The labour movement and women’s right movement facilitated social discussion and had a great impact on literature as well. In 1917, Finland was declared independent from Russia and in 1918 Finland went through a traumatic civil war. However, the European influences

were also important for the development of Finnish literature. Realism arrived from Scandinavia in the 1880s. In the 1920s, the *Tulenkantajat* (the Flame Bearers) advocated cosmopolitanism and sought to build international connections. Leftist writers came strongly forth in the 1930s. Economic depression and political crises briefly affected the number of literary publications in the 1930s, but the numbers started to raise already at the end of the decade. The end point of our project is the end of the Second World War (1945). This *terminus ad quem* is chosen based on two criteria. First, the Second World War marked a thematic break in Finnish literature. Second, in Finland literary works are released from copyright 70 years after the death of the author, meaning that much of the literature published since the 1940s is not in the public domain.

The main dataset of the project is the public domain *Projekti Lönnrot* corpus of digital books in Finnish from the nineteenth and early twentieth centuries. The corpus has been created by volunteers, who have corrected the spelling of the digitised books. Therefore the plain texts are mostly free of noise and other errors from Optical Character Recognition (OCR) scanning, i.e. misidentified letters. *Projekti Lönnrot* includes classic works of Finnish literature but also popular fiction and other more marginal genres. However, the corpus also contains translations from other languages and non-fiction books, which we have filtered out. This ready-to-use resource in public domain has provided an excellent starting point for our project.

The second corpus we use for the project is *Project Gutenberg*, which also contains transcribed Finnish fiction texts. Much of its work is included in *Projekti Lönnrot*, but there are also many supplementary texts: Only texts that are not in the Lönnrot collection have been manually selected. We have written a parser to segment the Gutenberg texts into chapters. After this step, the data has been processed with the same pipeline as the Lönnrot corpus.

1.3 Metadata

Collecting and curating the metadata has constituted a major part of the first year of our project. Collecting metadata consisted of five phases that were interlocked with the process of collecting the full texts:

1. In the case of *Project Gutenberg*, searching and selecting the Finnish works included in the multilingual collection. By contrast, the works in *Projekti Lönnrot* are all published in Finnish, although some are translated from other languages.
2. Excluding the translations. *Projekti Lönnrot* and *Project Gutenberg* include many Finnish translations of works written elsewhere and/or in other languages; if the translator is Finnish, the work has been included in these corpora, but we have excluded the translations.
3. Manually reviewing all the Finnish works to exclude non-fiction and to find the ones matching the time span of our project.
4. Solving the question of the identity of the writer. Should collectors and editors of folk poetry be regarded as writers, and to which extent? The use of pseudonyms also caused problems in the collection of metadata.

5. The division of edited collections into individual works. *Project Gutenberg* and *Projekti Lönnrot* include many volumes of collected works. Often the information about the original publishing date of each individual work was not easy or even possible to find. Many poems, dramas and short stories were originally published in newspapers or periodicals. In these cases, we have included all metadata that could be found (original publishing date, original title) in addition to the metadata of the edited collection.

2. Process Chain

2.1 Parsing Plain Text to TEI

The source files from the *Projekti Lönnrot* are plain text files, very similar to what the *Project Gutenberg* uses. The files are encoded by human volunteers. In the process of encoding, the texts are systematised to a certain extent, but as perhaps can be expected, the results are not fully systematical: *E.g.* book sections are usually indicated with a separator of four empty lines, but there can also be five. Moreover, the unit of digitisation is a physical volume which, for example, in case of collected works and anthologies, results in many individual works appearing in one *Projekti Lönnrot* item.

In addition, the resulting text files use various encoding systems, UTF-8, Win-1252 and ISO-8859-1 among others. This is a problem for Finnish texts, since various special characters (“ö”, “ä”, “å”, and their capitalised versions) are used that are encoded with different code points in each coding system. These letters are very common, and also significant: The meanings of the words “läski” and “laski”, for example, have nothing to do with each other. For processing the texts, it was necessary to transform everything to Unicode.

This was done by analysing each file with the Unix `file` utility.¹ The utility returns the guessed encoding for a file based on an analysis of coding points in the texts, and in most cases, the use of this reported encoding as argument to Java’s file input/output routines resulted in a correct read.

The actual conversion problem was connected with the aims of the whole project. The most simple way to process the texts would be to treat them as plain texts, tokenise them and run through relevant tools. This approach is often used when NLP tools are developed and measured, and the results are usually in some kind of standardised format, like CoNLL-U² or similar. The purpose of this project, however, is to use the NLP tools only as a first stage in a production context for geoparsing the texts and representing their geographical information on maps. We do not focus on these results as such, but use them further in working with the texts on a different level. Therefore, the CoNLL-U type format is not suitable for the project. Rather, we need to integrate the annotations produced by the NLP processes into XML documents that are more suited to both visual presentation of the text and dissemination of the annotated documents.

In order to fill the main obligations of the project, we decided that the texts resulting from the processing pipeline should be TEI-encoded XML files in order to facilitate their

1. `file -b -mime-encoding`.

2. See: <https://universaldependencies.org/format.html>.

further use. This suggested a possible approach where the texts were converted to TEI as early as possible. These texts were then to be used as sources for further analyses.

The conversion process was divided in several phases:

1. parse the text files;
2. convert to parse-based XML;
3. convert this XML to TEI.

The parser for the texts was written using the EBNF³ notation. The actual parse process was done with Clojure code, running the Instaparse library (Engelberg 2022).

The parse was then converted to an XML document with the same structure that was defined in the parser definition. This step is not strictly necessary, but producing TEI directly from the parse results is very complicated, requires a lot of manual coding and is therefore error-prone. A better solution is to export the parse results as XML, and then further process this XML with a suitable XSLT to TEI (`parse-to-tei.xsl` in the project repository).

As this is the step where the individual works in the source volumes are recognised (see the parser definition, especially the element `work1`, in the file `parser.bnf` in the project repository), this was the occasion to create individual ID's for these works. The IDs were created following a simple scheme: "`lonnrot_<basename>_<serial>`". At first, we have the string "`lonnrot`" or "`gutenberg`" to signify the data corpus, then the `basename` of the file which is the same as the volume ID in the *Projekti Lönnrot* or *Project Gutenberg* corpus, and as a third element, the serial number of the recognised `work1` element. This ID is stored as the `xml:id` attribute of the TEI element.

As a result of this conversion process, we have a set of files in TEI/XML format that corresponds very closely with the original *Projekti Lönnrot* files.

2.2 Splitting to Works and Adding Metadata

The problem with the first conversion process is that the resulting files closely resemble the structure of the original text files. They do not contain any metadata, which is difficult to extract reliably from the original files. Moreover, due to the structure of the parser, the structure of physical volumes that contain more than one "work" by an author or multiple authors are preserved. This means, that for these volumes, there is the main TEI document reflecting the actual volume, containing each work as separate, subordinate TEI documents.⁴

The separation between different works is marked with (at least) six empty lines in the *Projekti Lönnrot* files. A problem rises at the beginning of the digitised volumes, because the first work contained in the physical volume is often, but not always, separated from the header data by six empty lines. So for example a book containing a single work can have six empty lines between the book bibliographic data and the actual work, but not always; in many cases, the actual text begins after only four or five lines, which

3. Extended Backus-Naur form.

4. This is supported by and allowed for by the TEI specification.

is otherwise used to separate different chapters within the works. This problem is also present in volumes containing multiple works, in the form of a missing separator between the book header and the first work.

In practice, the main `tei` element can be the whole work or not; if it has one subordinate `tei` element, this can mean that there is one work in the volume, but there can be also two works, the first just being without a separator from the volume header. And whatever the number of the works, each has its own set of metadata (see [subsection 1.3](#)) that must be added to the actual work.

The external metadata offered a solution in the form of providing information about how many works a volume should contain. This information, combined with the ability to extract all `tei` elements from the documents and count them, made it possible to combine work metadata with the right works. The extracted `tei` elements were also turned into independent documents at this stage.

Another process that was implemented at this point was the identification of individual tokens in the text. The parsing process created individual tokens in the resulting data using the TEI's elements `w`, `pc` and `num`. Since the idea was to further process the texts with external algorithms, it was necessary to think in advance about how to merge the results back to the TEI documents, and for this reason, giving unique IDs to individual tokens was deemed necessary.

A simple naming scheme, based on the document ID (see above), was created:

`<document_id>_token<serial>` where the `document_id` was the same used in the `xml:id` of the TEI element, and the `<serial>` was calculated using the XSL `accumulator` element.

As a result of this process, we have each “work” in the dataset (novel, play, etc.) in its own TEI file, with a unique ID used both as the `xml:id` attribute of the TEI root element as well as the base name for the XML file. Each work has a basic set of metadata defined in the TEI document header, including the author, the title, and the year of publication, as well as statements regarding the production of these digital editions of the works.

Each work has its main internal divisions marked with the TEI `div` elements, and paragraphs are surrounded with the `p` tags. The texts are tokenised using the respective TEI elements. Spaces are not marked, but they are left in the XML text. Since the whitespace handling of XML can occasionally be tricky, and some tools may interfere with spaces, these can also be reconstructed later: The information about the lack of spacing is stored using the `join="left"` attribute in the cases where the token does not have any whitespace before it. Thus, each token now has an ID that is unique for the whole project corpus.

As a result of this process, at the time of writing, we have a dataset of 848 texts in Finnish language, published for the first time between 1870 and 1944. These texts include altogether 20,356,701 tokens ([Table 1](#)).

2.3 Named Entity Recognition and Lemmatisation

One of the key goals of the project is to automate the processes of place name recognition and referencing.

	1870s	1880s	1890s	1900s	1910s	1920s	1930s	1940s
fiction	216,925	1,795,264	2,422,107	3,115,430	4,703,275	3,638,502	1,201,012	595,485
drama	18,060	149,437	246,392	497,496	377,157	201,590	36,375	15,654
poetry	29,430	74,873	111,124	195,146	315,331	91,529	36,486	4,041
misc	0	0	0	22,683	108,038	115,939	21,920	0
TOTAL	264,415	2,019,574	2,779,623	3,830,755	5,503,801	4,047,560	1,295,793	615,180

Table 1: Sum of tokens per genre and decade.

Many of the projects with similar aims are using geotagging processes based on either recognising certain types of names or using a gazetteer. For example, the *Edinburgh Geoparser* uses the latter approach, and requires therefore a pre-defined set of place names to work with (Alex et al. 2019). An example of the former method is the retrieval of Parisian street names from French novels between 1800 and 1914 that was based on the typical structure of street and other place names in the French language, where for example “rue” forms a part of most street names when referred to in the text (Moncla et al. 2017, 2019).

Instead of trying to create our own heuristics for this, we chose a very different approach by using Natural Language Processing algorithms created by the TurkuNLP group⁵, especially the *Finnish NER* (Named Entity Recognition) tool (Luoma et al. 2020). The *Finnish NER* system analyses the source text using a natural language model, and recognises various types of named entities in the text, including geopolitical place names, natural place names, persons, buildings, organisations, etc.⁶

The system takes as its input either a plain text file, which it tokenises itself, or an already tokenised list of words and punctuation where each token is on its own line. The latter is the only option if the purpose is to somehow connect the results with the original data, for in this case, there is a strict correspondence between the lines in submitted source data and received results.

As the words, punctuation, and numbers in the source documents were already tokenised in the previous stage, it is a simple task to extract a part of the source document as lines where each line contains the token ID and the content, separated with a tab. From this data, the content column can be separated, given as argument to the *Finnish NER*, and the results can be merged back with the original extract containing the token ID’s. This list can then be used as source data for an XSLT merging selected data from the results with the original TEI files.

In practice, this means choosing selected entity types from the NER results, finding the token ID’s covered by each named entity, and then surrounding these elements in the TEI with the corresponding tag.

Initially, it seemed that the TEI elements denoting places would be appropriate to describe OntoNotesNE types (see Table 2 for correspondences between OntoNotesNE types and TEI elements). However, this mapping between the OntoNotesNE types produced by the *Finnish NER* (Table 3), and the entity types offered by TEI is not without problems, for the TEI categories do not fully correspond to the OntoNotesNE types. For

5. See: <https://turkunlp.org/>.

6. Full list of recognised entity types can be found at the system web pages, and further description of the entity types in the OntoNotes Manual (Weischedel et al. 2012, 21).

OntoNotesNE type	TEI element
GPE	<placeName>
LOC	<geogName>
PERS	<persName>
ORG	<orgName>

Table 2: Correspondence between the OntoNotesNE types and TEI elements.

OntoNotes entity types	Description
PERSON	People, including fictional
NORP	Nationalities or religious or political groups
FACILITY	Buildings, airports, highways, bridges, etc.
ORGANIZATION	Companies, agencies, institutions, etc.
GPE	Countries, cities, states
LOCATION	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Vehicles, weapons, foods, etc. (Not services)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK OF ART	Titles of books, songs, etc.
LAW	Named documents made into laws
LANGUAGE	Any named language

Table 3: The list of the OntoNotesNE types.

example, OntoNotesNE uses the FAC(ility) tag to denote buildings, roads, and other man-made architectural structures. In the TEI documentation, buildings are annotated under the name tag:

- I never fly from <name key="LHR" type="place">Heathrow Airport</name> to <name key="FR" type="place">France</name>

(TEI Consortium 2022, 13.1.1 Linking Names and Their Referents)

To be able to tag facilities, historical events, or nationalities in texts, we need to use the XML tag “name”, where the type of the entity is given in the “type” attribute. We are currently considering a solution to this problem that would be in line with TEI principles, but at this point we have decided to use the name tag because it is suitable for all tagged entities and still meets the TEI specification.

As a result of this process, we have TEI documents with place names marked with name tags. In addition to the actual tags marking the entity, each annotation was given a unique ID using the `xml:id` attribute so that they can be referred to from elsewhere, just like was done with the tokens in the earlier stage.

For the next phase, Named Entity Linking, we need to have the place names in a form that can be used as search string in geodatabases. The NER process used above uses a natural language model, and does not lemmatise the tokens in the process; consequently, the resulting place names are still in their inflected form. In the Finnish language, common and proper nouns decline in 14 cases. For example, the nominative case for Berlin in Finnish is “Berliini”, and when someone is in Berlin, the inflected word form is “Berliinissä” (inessive case). Whereas “Berliini” is the nominative form of the word, and can be found as such in many databases, “Berliinissä” usually does not return any results.

Therefore, the results also need to be lemmatised in order to recover their root forms. For this purpose, we use another system created by the TurkuNLP group, the *Turku Neural Parser Pipeline* (Kanerva et al. 2018, 2021) (TNPP). This system accepts the input in a similar format as the NER system, and returns results in the CoNLL-U format.

Since the input data for both systems is the same, the lemmatisation process was merged with the NER process. Because the results of the NER process had to be merged with the TEI files in four separate runs, the merging of lemmas just added a fifth data merge with XSLT, and as a result, the documents were lemmatised at the same time.

This is a side effect of using the TNPP system. Moreover, this system is based on the natural language model, and each token is analysed in the context of a sentence, where its position in relation to what comes before and after is significant. The entity names cannot therefore be extracted as individual names and lemmatised; but this also yields the lemmatisation of the whole documents. This POS (part of speech) data may yet prove to be useful in the later stages of the project, because we can link place names to their original context, including information such as which adjectives or verbs refer to them.

After running this process for the data files, we have a set of TEI document files, where in addition to the results of [subsection 2.2](#), each token has also the lemma of its content word, and the four name element types shown in [Table 2](#) are marked in the texts.

2.4 Named Entity Linking

After the Named Entity Recognition process, the various entity types recognised are annotated with the respective TEI elements in the documents. This does not yet take us much further on the path of geographical analysis of the texts; we might be able to create some statistics about the density of various entity references in the works by different authors or of various genres, but these annotations do not allow for any cartographic analyses of the documents.

The place names need to have some additional data, coming from outside of documents, in order to be usefully analysed in any wider context. In principle, there are two ways to approach the geographical linking of the place names in the documents.

In the first option, for each place name, some kind of geographic location is looked for. This could be as simple as a reference point, expressed in any coordinate system (practically geographical data in any coordinate system can be transformed to WGS84, which is the format assumed by TEI and supported by all libraries and applications for geographical analysis), which could be stored together with the place name in order to provide a geographical location. This location can then be used to create geographical presentations of the texts, either individually or in larger sets.

In the second option, for each place name, the corresponding *place* entity in some reference system is found. This place entity is what we generally mean when we talk about different places. For instance, “Berlin” is a place entity, but “52°31’N, 13°23’E” is a string containing the latitude and longitude in degrees of a point that could be used to represent the location referred to by the place name “Berlin” in the text. If instead of storing this string, we store a reference to the entity “Berlin, the Capital of Germany”,

we can use this entity to gather more information about the places referenced in the documents.

The second option also provides a way to control the results of the linking process. In the first option, what happens behind the scenes is obscured by storing only the results. Why is this particular location stored, and not some other? Without any other information than the geographical coordinates, it is not possible to understand why the reference in our text to “Berlin” suddenly shows up as a point in Wisconsin, USA, on our geographic display of the results. The second option is somewhat more complicated, but not too much not to be preferred.

For further processing of the data, another set of XSL transformation templates was created. In practice, each Named Entity Linking based on already recognised entity names uses some kind of gazetteer of entities. For geographical data, there is a myriad of options to choose from. No comprehensive comparison of these gazetteers was done at this time, but with some preliminary tests it was easy to gather that for example *Getty Thesaurus of Geographic Names*⁷ has limited coverage of the local place names that appear in the data. GeoNames⁸ has better coverage, and also an API that allows for more structured searches of the data. DBpedia Spotlight uses DBpedia for disambiguation and linking of named entities (Mendes et al. 2011). However, the Finnish version of DBpedia is currently not maintained.

In the first phase of this project, Wikidata was chosen as the geographical database, mainly because of existing knowledge about Wikidata’s data model and the use of SPARQL as a query language. Moreover, Wikidata has a good-quality search engine for searching the entities in the data based on their preferred and alternative labels and textual descriptions. This search also includes a ranking algorithm, which allows for the retrieval of the most probable result. In contrast, DBpedia Spotlight uses a ranking algorithm based on the Inverse Candidate Frequency (ICF) weight (Mendes et al. 2011, 3).

In the first stage of entity linking, each source document was processed for its place and location annotations. For each annotation, the place name it referred to was obtained in its basic form using the lemmatised data, along with the unique ID of the annotation (see above).

When figuring out the search string, another problem manifested. A typical form of place name that appears also in these texts is “Suomen Suuriruhtinaskunta” (“Grand Duchy of Finland”), which is composed of one or more genitive forms (“Suomen” is the genitive of “Suomi”) before the base word in nominative (“suuriruhtinaskunta”, “Grand Duchy”). The parts in genitive are not inflected, so the inessive form of the place is “Suomen Suuriruhtinaskunnassa”, etc. A lemmatiser, however, returns the lemmas of each part, so the lemmatised form of the name is “Suomi suuriruhtinaskunta”, which usually does not return any results from any geodatabase.

The POS (part of speech) data provided by the lemmatiser is valuable in this case. In the case the place annotation contains more than one word, if the words before the last were in genitive form, this form should be returned instead of the lemmatised nominative,

7. See: <https://www.getty.edu/research/tools/vocabularies/tgn/>.

8. See: <http://www.geonames.org/>.

which then should be used only for the last word in the multiword annotation. In this way, these type of multiword place annotations returned their referent in a form that actually is usable in searching the place data. This heuristic was coded as a function in the XSL transformations, and used in the data retrieval process.

For each name, the local cache was checked for corresponding content, and if no content was found, data was looked for in Wikidata using the SPARQL endpoint. A query template was defined for an efficient search of the place and location name labels in Finnish, limiting the results to those results that were also descendants of appropriate geopolitical and natural place types, respectively. The template for the geopolitical place name query is shown below (with the query term “York”).

```

1 PREFIX bd: <http://www.bigdata.com/rdf#>
2 PREFIX mwapi: <https://www.mediawiki.org/ontology#API/>
3 PREFIX wd: <http://www.wikidata.org/entity/>
4 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
5 PREFIX wikibase: <http://wikiba.se/ontology#>
6
7 SELECT distinct ?place ?placeLabel ?country ?countryLabel ?location ?
      geonamesID ?openstreetmaprelid ?description WHERE {
8   service wikibase:mwapi {
9     bd:serviceParam wikibase:endpoint "www.wikidata.org";
10      wikibase:api "EntitySearch" ;
11      mwapi:search "York" ;
12      mwapi:language "fi" .
13      ?place wikibase:apiOutputItem mwapi:item .
14      ?num wikibase:apiOrdinal true .
15   }
16   ?place wdt:P31/wdt:P279+ wd:Q56061 .
17   optional {
18     ?place wdt:P17 ?country .
19   }
20   optional {
21     ?place wdt:P625 ?location .
22   }
23   optional {
24     ?place wdt:P1566 ?geonamesID .
25   }
26   optional {
27     ?place wdt:P402 ?openstreetmaprelid .
28   }
29   service wikibase:label {
30     bd:serviceParam wikibase:language "fi" .
31   }
32   optional {
33     ?place schema:description ?description .
34     filter (lang(?description)="fi")
35   }

```

```

36 } order by ?num
37 LIMIT 1

```

When adding the returned place and location information to the local cache data, the place is given an ID that can be used to refer to it. Repeating this process for all place and location names gathered from the texts created a list of annotation ID's and place ID's that was used to modify the TEI files by adding ref-attributes pointing to the place ID's to each place and location annotation. The contents of the cache were stored as records in a MongoDB-database, so the list can be reconstructed and replaced later.

The TEI files were then updated with the retrieved reference data. The ref-attributes of each annotation element were updated with content that can later be used to link each annotation to the place record in the MongoDB database.

2.5 Resulting Dataset

Once the text has gone through all the steps in the process, an annotated sentence looks like this:

```

1 <p><pc xml:id="lonnrot-0585-1-token3137" n="3137">-</pc><pc xml:id="
  lonnrot-0585-1-token3138" n="3138">-</pc> <w xml:id="lonnrot-0585-1-
  token3139" n="3139" lemma="pikkunen" pos="ADJ" msd="Case=Nom|Degree=
  Pos|Derivation=Lainen|Number=Sing|Style=Coll">Pikkunen</w> <w xml:id="
  lonnrot-0585-1-token3140" n="3140" lemma="käärö" pos="NOUN" msd="Case=
  Nom|Number=Sing">käärö</w> <w xml:id="lonnrot-0585-1-token3141" n="
  3141" lemma="olla" pos="AUX" msd="Mood=Ind|Number=Sing|Person=2|Tense=
  Past|VerbForm=Fin|Voice=Act">olit</w><pc xml:id="lonnrot-0585-1-
  token3142" n="3142">,</pc> <w xml:id="lonnrot-0585-1-token3143" n="
  3143" lemma="kun" pos="SCONJ" msd="_">kun</w> <w xml:id="lonnrot
  -0585-1-token3144" n="3144" lemma="ankara" pos="ADJ" msd="Case=Nom|
  Degree=Pos|Number=Sing">ankara</w> <name key="/PERSON/
  PERSON_Heblarouva?lemma=Heblarouva" type="PERSON" xml:id="lonnrot
  -0585-1-annotation-PERSON-101"><w xml:id="lonnrot-0585-1-token3145" n=
  "3145" lemma="Hebla#rouva" pos="NOUN" msd="Case=Nom|Number=Sing">Hebla
  -rouva</w></name><pc xml:id="lonnrot-0585-1-token3146" n="3146">,</pc>
  <w xml:id="lonnrot-0585-1-token3147" n="3147" lemma="iso#äiti" pos="
  NOUN" msd="Case=Nom|Number=Sing|Number[psor]=Sing|Person[psor]=2">
  isoäitisi</w><pc xml:id="lonnrot-0585-1-token3148" n="3148">,</pc> <w
  xml:id="lonnrot-0585-1-token3149" n="3149" lemma="viedä" pos="VERB"
  msd="Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act">
  vei</w> <w xml:id="lonnrot-0585-1-token3150" n="3150" lemma="sinä" pos
  ="PRON" msd="Case=Acc|Number=Sing|Person=2|PronType=Prs">sinut</w><lb
  /> <w xml:id="lonnrot-0585-1-token3151" n="3151" lemma="täältä" pos="
  ADV" msd="_">täältä</w> <w xml:id="lonnrot-0585-1-token3152" n="3152"
  lemma="koti" pos="NOUN" msd="Case=Ill|Number=Sing|Person[psor]=3">
  kotiinsa</w> <name key="/GPE/GPE_Siuntio?lemma=Siuntio&wikidata_id
  =Q984931" type="GPE" xml:id="lonnrot-0585-1-annotation-GPE-27"><w

```

```
xml:id="lonnrot-0585-1-token3153" n="3153" lemma="Siuntio" pos="PROPN"
msd="Case=Ill|Number=Sing">Siuntioon</w></name><pc xml:id="lonnrot
-0585-1-token3154" n="3154">.</pc><lb/><lb/></p>
```

The sample cannot be prettyfied, since the contents of the <p> element is a mixed-mode XML, where the whitespace between different elements is semantically important.

The sentence contains a reference to the GPE entity “Siuntio”, which is linked to the Wikidata record Q984931. The key tags contain the information needed by the web front end to link to the entity. The sample shows how each element has been given an ID that is unique to the whole project dataset. Also, the discovered annotations, marked with <name> elements, have unique ID’s.

3. Metachain

3.1 Managing the Workflows

When working with datasets containing more than some tens of data units, managing the workflows becomes a challenge.

A division into three different types of workflow management is proposed here:

1. Running different algorithms and analysis tools individually on each unit of data in the dataset;
2. Writing scripts to either chain many stages of the analysis in order to run this whole chain for each data unit, running each stage for numerous data units, or as an end product, to run many stages for all data units;
3. Manage the running of each stage and data unit with some external tool, only triggering the required phases for relevant data units.

The first type of workflow management is usually the one chosen when experimenting with new tools and datasets. Data units are analysed individually using a graphical user interface, or with shell commands. This is manageable for a small number of data units, but increasing the amount of data and with repeated applications of the tools, this soon becomes very cumbersome.

The usual solution is to turn the management process into a script, and let the script run the analyses; the script is run on external services, with more efficient servers or cloud instances. This adopts the second type of workflow management.

This process tends to end up in difficulties when used in digital humanities. The script-based workflow management is uncomplicated to use only in the cases where the source data is “born digital”, meaning, it is originally produced as digital data and therefore corresponds to a well-defined, strict data scheme. We have in this paper already encountered a situation where the data was not what it was expected to be: The *Projekti Lönnrot* corpus texts use different encoding systems, and there is no systematic information anywhere about which file uses which encoding. In fact, the situation is even worse: There are files that do not fully follow *any* known encoding. Often this kind of situation is caused by only a single character in the file, but this kind of error in file

reading usually causes the program processing the file to throw an exception, if this has been anticipated, or otherwise, to just crash.

This kind of crash has to be managed somehow. The faulty file has to be recorded somewhere, and it has to be decided, in advance, how to continue the process. Further, there has to be a way to manage the metadata so that if only the faulty file is skipped, there is a way to retrieve a list of the faulty files, and also, when these are fixed, not necessarily to process the whole dataset again, but only the ones that are missing. This kind of scripting becomes quickly very complicated, and the superficially rather simple script-based approach starts to acquire features that actually are already present in other kinds of systems, which brings us to the third kind of workflow management.

This approach to project workflows is more of a methodology rather than a straightforward solution, but the fundamental concept is straightforward: utilise a system that can automate processing chains, while also keeping track of results and errors, to manage the various phases and units of data within a project.

In our case, the management of the processing chains was created around *asynchronous message queues*. Each individual stage in the processing chains is devised as an independent client, that both asks relevant queues for work to do and sends either results or metadata about completed works to another queue. In relevant terminology, each tool functions as a consumer, retrieving work from designated queues and publishing completed results or potential errors to the appropriate queues.

The queues are managed by a *RabbitMQ* message broker, which is a simple yet powerful tool for this purpose. It has a satisfactory browser-based user interface, which facilitates the monitoring of the progress of various queues, as well as the examination of the contents of queues, especially in instances where errors have been reported.

The actual clients can reside anywhere where it is possible to access the message broker. This is beneficial because it enables the utilisation of remote resources for data processing. For example, in our case, the main data servers are located on the local university premises, but some CPU-intensive processes are run on virtual machines at the national computational resource provider.

3.2 Manual Intervention and Human Contribution

Luoma et al. (2020) propose that their Finnish NER is capable of reaching around 90% of precision and recall for results in texts drawn from most domains. However, they have not assessed the performance of the NER tagger using fiction from the 1870s to the 1940s. It is possible to distinguish false positives by checking the statistics of the NER results, although some cases require going back to the annotated text to evaluate the context in which the annotation belongs. But the only way to ensure how many spatial entities NER has failed to identify is to read the texts and check the annotations one by one. The manual review of the results is time-consuming even if only a randomly sampled portion of them is checked. The NER results are currently under review in our project, but the preliminary review of the results indicate that they are of sufficient quality.

Text	Precision	Recall	F1-value
Santeri Ivalo: <i>Anna Fleming</i> (1898)	0.92	0.97	0.94
Algot Untola: <i>Kuolleista herännyt</i> (1916)	0.93	0.84	0.88

Table 4: Precision, Recall, and F1-values calculated for geopolitical place name recognition in manually controlled texts: Santeri Ivalo’s historical novel *Anna Fleming* from 1898 and Algot Untola’s novel *Kuolleista herännyt* from 1916.

Text	Precision	Recall	F1-value
Santeri Ivalo: <i>Anna Fleming</i> (1898)	0.97	0.89	0.93
<i>Anna Fleming</i> (with NEL added) (1898)	0.81	0.61	0.70
Algot Untola: <i>Kuolleista herännyt</i> (1916)	0.33	0.5	0.40

Table 5: Precision, Recall, and F1-values calculated for geographical place name recognition in manually controlled texts: Santeri Ivalo’s historical novel *Anna Fleming* from 1898 and Algot Untola’s novel *Kuolleista herännyt* from 1916. For *Anna Fleming*, we also include respective values for the combined NER & NEL process.

Two of the texts were manually controlled by the project members. In addition to checking the recognised entities, the missing cases were also calculated, allowing us to compute valid precision and recall values, as well as the F1-values. (Table 4 and Table 5.)

The Finnish language of the late nineteenth century may pose more challenges for the NER algorithm, as the Finnish literary language was still taking shape at that time. We will now look at some problematic cases, using as an example the historical novel *Anna Fleming*, published 1898 by Santeri Ivalo. The novel, rich in geographical references, is set in Sweden in the sixteenth and seventeenth centuries and, to a large extent, in the area now known as Finland. The main characters in the novel travel to places, reminisce over their stay in different places, talk about travelling to different locations and are related to people who live in different parts of Europe. The NER algorithm has worked very well for this challenging text, but it makes certain systematic errors. There are some examples of a classification error, where a political entity is tagged as a geographical place or vice versa. Most mistakes were found with the names of medieval manors like Kuitia, Liuksiala and Kankainen. For example, Kuitia (a manor house founded in the fifteenth century in Southwest Finland) is mentioned 11 times in the first chapter of the book. It is tagged as a geographical formation six times, as a geopolitical entity (which is the correct tag) two times and as a person once. On four occasions, the place has been completely unmarked.

The names of historical or geographical entities that no longer exist also sometimes cause problems for the tagger. For example, “Danzig” (modern Gdańsk), is in the second chapter of the book tagged as a person, and so is “Iharinkoski” (Ihari rapids) in the third chapter. On the other hand, tagging geographical places usually works fine: “Vantaankoski” (“Vantaa rapids”), “Suomenlahti” (“Gulf of Finland”) and many others were all tagged correctly, even if the names are Swedish ones like “Sandö” or “Estnäs”. Cities, provinces, or countries were usually tagged correctly. For example, “Suomi” (“Finland”) occurs in the text in inflected forms “Suomessa”, “Suomen”, “Suomea”, “Suomesta” and “Suomeen” but it is nevertheless correctly tagged every time (for example, 50 times in the second chapter).

Similar inaccuracies are present also in the novel *Kuolleista herännyt* (*Risen from the Dead*, 1916) by Algot Untola, who is probably better known by his pseudonym Maiju Lassila. In the novel, an uncultured and illiterate dockworker, Jönni Lumperi from Helsinki, gains 2,000 marks from a lottery and is encouraged by a wealthy businessman into an endeavour to turn the sum into millions. Jönni chases this dream of enrichment around Southern Finland, first to Tampere and its rural surroundings, then to Hämeenlinna and Riihimäki. The journey ends back to Helsinki with all the lottery winnings and more lost. The novel contains many geopolitical place names (GPE) and some natural locations (LOC) both real and fictional. NER has recognised many of these quite well, but some inaccuracies also occur. Manual proofing of GPE and LOC tags by NER has revealed 60 clear errors (the novel contains altogether 213 GPE and LOC tags). Some of them are most likely due to outdated spelling of place names, such as “Marseljeesi” (“Marseille” in current Finnish spelling) or “Söörnäinen” (nowadays “Sörnäinen”, a neighbourhood in Helsinki). Others errors are fictional non-GPE locations and biblical place names such as “Kaana” (“Cana”) etc. Sometimes person names are recognised as non-GPE locations and nouns as proper names. There is also a surprisingly high number of cases when Helsinki and Tampere were not recognised in a declined form even though the spelling is modern, and both are large and well-known cities in Finland. The errors do not seem systematic, and both cities are more often correctly recognised than not.

On the whole, inaccuracies are quite rare and in most cases do not repeat. Yet, it is worthwhile also to point out two cases of inaccuracies of a special nature:

1. The chapter 17 introduces a new character named “Hesa”. This person name (PERS) is tagged 13 times out of 21 as GPE location, most likely due to the frequent modern use of “Hesa” as the nickname for Helsinki. In three cases when “Hesa” is followed by the family name “Ruokka” it is recognised as a person name and only in one case without the family name. Also, in four cases “Hesa” is not tagged at all, one of them followed by the family name. The “Hesa” case exemplifies that NER is capable to recognise even nicknames for places, but also that from this fact can follow unpredictable inaccuracies. Had the main character been named “Hesa”, hundreds of false positives would have occurred.
2. A more peculiar case to explore in more detail is the Estate of “Punturi”. This estate is referred to in many ways: “estate”, “house”, “farm” and “land of Punturi”. In this case, the ambiguity of natural language makes it very hard to even control if the name is tagged correctly. As the name of the estate is the same as its owners, even a close reading of the context does not always give a clear answer what is being referred to, the proper name of a specific land area, or the fact that a person named Punturi is an owner of a land area left unnamed. From 16 cases with reference to the estate, 10 are recognised as GPE and 6 cases are not tagged at all. In some cases, it is very clear that reference is to the name of the land, and it should be tagged as GPE, but many ambiguous cases also occur. However, as previously stated, even manual proofreading and close reading cannot provide a binary classification, as natural language does not function in this manner. It should be noted that as our project investigates fictional literature, which is by nature meant to be open to multiple interpretations, these ambiguities are an

essential part of data material and should not be ignored, excluded or forced into binary categories.

4. Conclusion

In this article, we have described the process of converting plain text fictional texts into TEI XML format. The pipeline is fully automated as far as possible. The compilation and harmonisation of the metadata of the texts and the curation of the corpus has been carried out by a literary scholar with domain knowledge of the Finnish literary history of the period. The text corpus is then passed through a pipeline, which translates it into TEI XML format, segments it into separate chapters, lemmatises the text, searches for named entities and disambiguates and geocodes the spatial entities by linking them to open linked data.

The results of the NEL process are then checked by human readers familiar with the literature and culture of the period to assess the success of the identification. This cannot be done for the whole corpus, but requires the examination of randomly selected samples of works from different periods and by different authors. Although the basic idea behind our project – to put texts on maps – is simple, the Finnish-language corpus presents challenges: Named-entity recognition in Finnish has been difficult to implement with sufficient accuracy in the past. *Finnish NER's* ability to recall different names in text and the precision of the recognition are over 90% on contemporary Finnish. However, *Finnish NER* has been trained on modern textual material, so its accuracy is unlikely to be as good with nineteenth-century texts. Yet, the language model can be tailored to historical data using transfer learning (Labusch et al. 2019). We can experiment with this at a later stage of the project once we have enough human-corrected data available.

Finally, we will import the texts to an online interface where they can be read while exploring the maps drawn by the works and comparing the spatial regions of the different texts. As our project will produce a clean and structured TEI XML corpus of the texts, further scholarly examination of the data using other digital humanities methods will be convenient. For example, we can apply topic modelling or network analysis, as the corpus is already pre-processed by tokenisation, part of speech tagging, lemmatisation, etc. This allows us to extract the semantic features of the texts and associate them with the geocoded toponyms.

Our future plans also include studying and comparing Swedish-language Finnish fiction with Finnish-language fiction. Since the software we are developing is modular, adding Swedish name recognition to the pipeline will not be difficult. Moreover, due to its modular architecture, our pipeline, which has been released under an open licence, is also suitable for processing data in other languages. Thus, our work could be adapted to research and web publishing use with other lower-resourced European languages with relevant literary traditions.

5. Data Availability

The dataset for this study is available on Zenodo, see <https://doi.org/10.5281/zenodo.8365866>.

6. Software Availability

The code for running the pipeline is published at <https://github.com/harrikoo/extract-georef-finlit/tree/v1.0.0> and also stored on Zenodo at <https://doi.org/10.5281/zenodo.8369648>.

7. Acknowledgements

The 2-year project *Atlas of Finnish Literature 1870–1940* is funded by the Alfred Kordelin Foundation under the Major Cultural Projects Programme (2022–2024).

Computational resources were provided by University of Turku and CSC – IT Centre for Science, Espoo, Finland.

8. Author Contributions

Harri Kiiskinen: Conceptualization, Writing – original draft (main author), Data curation, Software

Asko Nivala: Writing – original draft, Project supervision

Jasmine Westerlund: Writing – original draft, Metadata curation, Data verification

Juhana Saarelainen: Writing – original draft, Data verification

References

- Alex, Beatrice, Claire Grover, Richard Tobin, and Jon Oberlander (2019). “Geoparsing Historical and Contemporary Literary Text Set in the City of Edinburgh”. In: *Language Resources and Evaluation* 53 (4), 651–675. [10.1007/s10579-019-09443-x](https://doi.org/10.1007/s10579-019-09443-x).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint*. [10.48550/arXiv.1810.04805](https://arxiv.org/abs/1810.04805).
- Engelberg, Mark (2022). *Instaparse*. Version 1.4.12. <https://github.com/Engelberg/instaparse> (visited on 12/06/2023).
- Frontini, Francesca, Carmen Brando, Joanna Byszuk, Ioana Galleron, Diana Santos, and Ranka Stanković (2020). “Named Entity Recognition for Distant Reading in ELTeC”. In: *CLARIN Annual Conference 2020*. <https://hal.science/hal-03160438> (visited on 12/06/2023).
- Gregory, Ian N., Christopher Donaldson, Andrew Hardie, and Paul Rayson (2019). “Modeling Space in Historical Texts”. In: *The Shape of Data in the Digital Humanities*. Ed. by Julia Flanders and Fotis Jannidis. Routledge. Chap. 5, 133–149. [10.4324/9781315552941](https://doi.org/10.4324/9781315552941).
- Gregory, Ian N. and Andrew Hardie (2011). “Visual GISTing: Bringing Together Corpus Linguistics and Geographical Information Systems”. In: *Literary and Linguistic Computing* 26 (3), 297–314.

- Kanerva, Jenna, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski (2018). “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*, 133–142. [10.18653/v1/K18-2013](https://doi.org/10.18653/v1/K18-2013).
- Kanerva, Jenna, Filip Ginter, and Tapio Salakoski (2021). “Universal Lemmatizer: A Sequence to Sequence Model for Lemmatizing Universal Dependencies Treebanks”. In: *Natural Language Engineering* 27 (5), 545–574. [10.1017/S1351324920000224](https://doi.org/10.1017/S1351324920000224).
- Labusch, Kai, Clemens Neudecker, and David Zellhöfer (2019). “BERT for Named Entity Recognition in Contemporary and Historical German”. In: *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*. https://konvens.org/proceedings/2019/papers/KONVENS2019_paper_4.pdf (visited on 12/06/2023).
- Luoma, Jouni, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo (2020). “A Broad-coverage Corpus for Finnish Named Entity Recognition”. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC2020)*. European Language Resources Association, 4615–4624. <https://aclanthology.org/2020.lrec-1.567> (visited on 12/06/2023).
- Mendes, Pablo N, Max Jakob, Andrés García-Silva, and Christian Bizer (2011). “DBpedia Spotlight: Shedding Light on the Web of Documents”. In: *Proceedings of the 7th International Conference on Semantic Systems*, 1–8. [10.1145/2063518.2063519](https://doi.org/10.1145/2063518.2063519).
- Moncla, Ludovic, Mauro Gaio, Thierry Joliveau, and Yves-François Le Lay (2017). “Automated Geoparsing of Paris Street Names in 19th Century Novels”. In: *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*. [10.1145/3149858.3149859](https://doi.org/10.1145/3149858.3149859).
- Moncla, Ludovic, Mauro Gaio, Thierry Joliveau, Yves-François Le Lay, Noémie Boeglin, and Pierre-Olivier Mazagol (2019). “Mapping Urban Fingerprints of Odonyms Automatically Extracted from French Novels”. In: *International Journal of Geographical Information Science* 33 (12), 2477–2497. [10.1080/13658816.2019.1584804](https://doi.org/10.1080/13658816.2019.1584804).
- Moretti, Franco (1998). *Atlas of the European Novel, 1800-1900*. Verso.
- Piatti, Barbara (2008). *Die Geographie der Literatur: Schauplätze, Handlungsräume, Raumphantasien*. Wallstein.
- Tally, Robert (2018). *Topophrenia: Place, Narrative, and the Spatial Imagination*. Indiana University Press.
- TEI Consortium (2022). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium. <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ND.html> (visited on 12/06/2023).
- Weischedel, Ralph, Sameer Pradhan, Lance Ramswah, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Jena D. Hwang, Claire Bonial, Jinho Choi, Aous Mansouri, Maha Foster, Abdel-aati Hawwary, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and Ann Houston (Sept. 28, 2012). *OntoNotes Release 5.0. with OntoNotes DB Tool co.999 beta*. <https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf> (visited on 12/06/2023).