

Enriching Cultural Heritage Knowledge Graph Metadata from Finnish Texts with Large Language Models

Rafael Leal^{1,*}, Annastiina Ahola¹ and Eero Hyvönen^{1,2}

¹*Aalto University, Dept. of Computer Science, Semantic Computing Research Group (SeCo), Finland, <https://seco.cs.aalto.fi>*

²*University of Helsinki, Helsinki Centre for Digital Humanities (HELDIG), Finland*

Abstract

This paper introduces the Finnish Named Entity Linker (FiNEL), a tool that leverages Deep Learning models, including Large Language Models (LLM), to recognize, disambiguate, and link Named Entities in Cultural Heritage texts. FiNEL is designed to enhance the metadata of textual documents by connecting them to Knowledge Graphs (KG). We propose a zero-shot classification method that resembles Retrieval-Augmented Generation (RAG) and discuss a prototype web service with a user interface that enables human intervention for final disambiguation decisions. This editing capability is crucial, particularly when automatic linking may be hindered by errors and hallucinations inherent in LLM-based tools. The paper also reflects on lessons learned from using FiNEL in applications targeting Digital Humanities (DH) research. Since the focus is on Finnish texts, our methods accommodate the specific challenges posed by this highly inflectional language and the available processing resources. Preliminary evaluation results underscore the potential of FiNEL: our named entity lemmatizer achieved an accuracy of 96.5% on the test dataset, while an LLM from the Llama family reached 97% accuracy for entities with only one candidate. However, accuracy decreased with each additional candidate.

Keywords

Large Language Models, Named Entity Recognition, Named Entity Linking, Linked data, Knowledge organization system, Knowledge graph

1. Introduction

Much of the data that could be used in Digital Humanities (DH) research is available only in unstructured textual form. Information extraction is then needed for creating metadata based on Knowledge Organization Systems (KOS) and Knowledge Graphs (KG) (Martinez-Rodriguez, Hogan, and Lopez-Arevalo 2020), publishing Linked Data Services, and building applications on top of them, such as the Sampo systems (Hyvönen 2022). For example, in our work on publishing the plenary session speeches of the Parliament of Finland as Linked Open Data (LOD) (Hyvönen et al. 2024), the speeches had to be linked to various domain-specific ontologies based on named entities (people, places, organizations, etc.), keyword resources, and a library classification system (Tamper et al. 2022). A fundamental task here is Named Entity Recognition (NER) and Linking (NEL). This paper addresses the question of how Large Language Models (LLM) can be exploited for the task where semantic disambiguation is a key challenge. This work is focused on Finnish texts, and we discuss some of the pitfalls that occur when performing natural language processing in this language. However, the ideas presented can also be applied to other languages.

The task of Named Entity Linking refers to the association of named entities in a text with their corresponding entries in knowledge bases. Knowledge bases store structured information about entities, using meaningful predicates such as "married to" or "founded in". The task is thus about bridging the unstructured text format with the structure of Knowledge Bases.

The 9th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2025), March 5–7, 2025, Tartu, Estonia.

*Corresponding author.

✉ rafael.leal@aalto.fi (R. Leal); annastiina.ahola@aalto.fi (A. Ahola); eero.hyvonen@aalto.fi (E. Hyvönen)

🌐 <https://seco.cs.aalto.fi/u/eahyvone> (E. Hyvönen)

📄 0000-0001-7266-2036 (R. Leal); 0009-0008-6369-4712 (A. Ahola); 0000-0003-1695-5840 (E. Hyvönen)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 Digital Humanities in the Nordic and Baltic Countries Publications – ISSN: 2704-1441

This task has a double purpose: on the one hand, it helps to describe and characterize texts by highlighting its themes, actors, locations and other relevant elements. By providing a Uniform Resource Identifier (URI) for each of those elements, the text itself becomes more computer-friendly, more transparent and less ambiguous. On the other hand, this process also allows for clustering, classifying and searching for specific information inside a dataset. Linking is a two-way relation that not only clarifies the contents of a text but also helps to place it in relation to other texts via its entities.

1.1. Motivation

The main objective of our work is to create a named entity linker that can contribute to increasing the amount of metadata available for Finnish unstructured texts. This means that Finns can work in their main language, Finnish, without having to rely on other languages and their tools to enrich and analyze metadata in applications for Digital Humanities. Although the main purpose of this tool is to enrich Cultural Heritage texts, it can also be used in other contexts. Thus, the linker benefits from being database-agnostic, so that any appropriate Linked Data database can be used as the target of the linking.

It is also important to note that we strive to be aware of the computational costs needed to run the system. In many cases, we chose much smaller, specialized local models even when calls to an external LLM could be carried out. Additionally, we aim at using open tools and models wherever possible. In the case of LLMs, local open-weight models are preferred instead of closed-source APIs. More transparent models would naturally be an even better fit for this project, but the scarcity of open models was a formidable enough challenge that made explainability take a step back. On the other hand, the nature of this system makes incorporating new models a straightforward process.

2. Related works

Traditionally, Named Entity Recognition and Named Entity Linking have been separated into different tasks, although newer works, capitalizing on the breakthroughs of deep neural networks, focus on end-to-end linking. For example, Ayoola et al. (2022) perform entity detection and disambiguation in a single forward pass, while Logeswaran et al. (2019) created a zero-shot linking system focused on addressing domain-shifting, without the need for gazeteers.

With the advent of LLMs, the field of Retrieval-Augmented Generation (Lewis et al. 2020) expanded dramatically, as summarized in Fan et al. (2024). The interface between LLMs and knowledge graphs can be considered a subdomain of RAG, and there are many recent works in this field: Baek, Aji, and Saffari (2023) created an LLM-based framework to answer questions by verbalizing triples related to entities in knowledge graphs, while Edge et al. (2024) used abstractive summarization over an entire corpus to find answers to global questions.

Regarding the Finnish language, Mäkelä (2014) described a web-based tool for annotating texts based on linked data, including named entities. Luoma et al. (2020) released a corpus and tool for Finnish Named Entity Recognition, expanding on Ruokolainen et al. (2020). This work has been used for example for pseudonymization of court documents (Oksanen et al. 2019) as well as for studying Parliamentary data (Tamper et al. 2022). However, as far as we are aware, there are no previous works using neural network techniques for entity linking in Finnish as in this paper.

3. Methods

This system follows the traditional three-step linking process: first, the named entities are identified; then, candidates are created for each of them; and finally, the entity is linked to one of the candidates. To this process model, we add a fourth step: a User Interface (UI) for manual revision and editing of the results. This is to ensure that, especially in critical contexts, such as speeches of parliamentarians, the linking of the entities mentioned in the text is accurate. The four steps are shown in Figure 1 and are explained in more detail below.

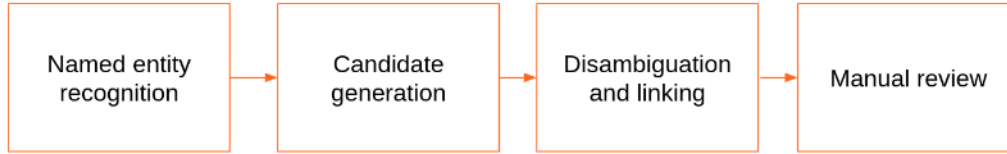


Figure 1: The four steps in the Entity Linking process.

3.1. Named Entity Recognition

The process of identifying named entities in a text is delegated to the third-party Named Entity Recognition tool created by the TurkuNLP Research Group (Luoma et al. 2020). This tool, which is a fine-tuned version of FinBERT and contains around 125 million parameters, outputs each word of the input text alongside a tag to indicate (1) if the word is part of a named entity or not and (2) which kind of entity this is, for example, a Person, a Location, or an Event. The list of entity classes is extensive, but our system reduces it to six: Person, Location (which groups the original Location, Facilities, and GPE categories), Event, Artwork, Date, and Organization. Any entity outside of these categories is ignored. As an example, an unstructured text might mention the entity "Michael Jordan". In the first step, the system should identify this as an entity and classify it as a Person.

An extra step that has to be carried out by our system is the lemmatization of the entities. This is a non-trivial task, since the Finnish language has about 15 cases and named entities are often multi-word phrases that should be lemmatized to different cases. For example, the phrase "Helsingin Sanomissa" means "in Helsingin Sanomat", in reference to a leading Finnish newspaper. Lemmatization, if done naively and word-by-word, will output "Helsinki sanoma" or roughly "dispatch Helsinki", ignoring the meaning and making it hard to identify the entity. The correct lemmatization, meanwhile, preserves the genitive case of the first term "Helsingin" and the plural case of the second, so that "Sanomissa" becomes "Sanomat".

We fine-tuned a Finnish T5 model `Finnish-NLP/t5-small-nl24-finnish`¹ that has around 260 million parameters, using a dataset of around 1 million Wikipedia internal links in Finnish, which we curated for this purpose. Internal links are those that associate an element from one page with another Wikipedia page, as can be seen at the end of the following sentence: "Vaikka Amsterdam on Alankomaiden perustuslain mukaan maan pääkaupunki, sijaitsevat [...] ulkomaiden diplomaattiset edustustot [[Haag]]issa."² The last word, "Haagissa", rendered as the element "[[Haag]]issa" in the the internal wikitext markup, is a hyperlink from the to the "Haag" Wikipedia page³.

This entity recognition step could be carried out using an LLM, but we have not evaluated their possible output. It is important to point out that the models we use in this step have several times fewer parameters than a typical LLM. For example, even the small local model Llama3.1-8B has a parameter count that is around 20 times larger than both models used in this step combined. Our model is much less computationally demanding and easier to use without specialized hardware due to the choice of model. And since the results are stored in a SQLite database (Hipp, Kennedy, and Mistachkin 2024), they can be easily ported elsewhere.

3.2. Candidate generation

The task of the candidate generator is to propose linking targets in a Knowledge Graph for the named entities. Taking advantage of the core strengths of Semantic Web technologies, candidates can originate

¹<https://huggingface.co/Finnish-NLP/t5-small-nl24-finnish>

²Text from url<https://fi.wikipedia.org/wiki/Amsterdam>, accessed 15.6.2025

³<https://fi.wikipedia.org/wiki/Haag>

from any Linked Data ontology, since they are identified by a unique Uniform Resource Identifier (URI). However, in order to be useful, candidates have to contain enough information about themselves so that the entity can be later disambiguated and linked to the most accurate candidate. In the example we gave above, a myriad of candidates can be created for the entity "Michael Jordan", since this is a common name in English-speaking countries: it might correspond to a footballer, a politician, a racing driver, a researcher, a basketball player, or other possibilities – even a song. Thus, candidates should contain a description of themselves, including their most relevant information points, to be later fed to the LLM in order to produce well-reasoned conclusions.

In our system, candidate generation is carried out via plugins. This architecture allows for changing the number and the priority of ontologies to be used depending on the project at hand, so that the targets of the linking are adjusted accordingly. FINEL goes through each plugin in order, stopping when suitable candidates are found for the entity in question. We have created generic plugins that use Wikipedia or Wikidata as ontologies, and plugins tailored to specific ontologies. For example, in the LetterSampo system, which is focused on searching, browsing, and analyzing correspondences between Finnish persons of the 19th century (Hyvönen et al. 2025), part of the entities had already been identified for some of the letters. These entities were enriched via links to Wikidata and made into their own plugin, which is given priority when searching for candidates. So, for each letter, these previously identified entities are searched first.

Each plugin can implement the search in different ways, but the generic process has candidate generation carried out against titles and aliases of entities using SQLite’s FTS5 search plugin⁴ in its default form. FTS5 provides full-text search on the contents of the database by tokenizing the relevant tables and creating an index of the location of each token, which results in fast queries. The results of the query are ranked via the BM25 algorithm inbuilt in SQLite.

The generic plugins we have created can be used with any text. They use arguably the most common linking targets in use, since these are open, permissive, robust and ever-evolving collaborative databases: Wikipedia⁵ and Wikidata⁶, both hosted by the Wikimedia Foundation⁷. The centrality of Wikipedia as a linking target is not a new phenomenon: this was already clear almost 20 years ago, when the term "wikification" was coined for this very specific task (Mihalcea and Csomai 2007). Wikipedia provides encyclopedic definitions for topics and entities, while Wikidata is a knowledge graph structured database created in 2012. Both offer open access and can be edited by its users.

3.2.1. Document memory

A virtual, in-memory SQLite table is created for each document. This table stores the names and aliases of entities that have been found in the document. When it comes to candidate generation, the system always searches this table first. This way, candidates that have been already seen can be found again if mentioned for example by alias or surname. In our example, “Jordan” would first recover the previously found “Michael Jordan”.

3.2.2. Finnish Wikipedia plugin

The Finnish Wikipedia plugin extracts information from Wikipedia dumps and Wikidata and transforms them into a SQLite database. The so-called "dumps" are downloadable copies of the contents of the wiki⁸. The first step is the identification of entities belonging to one of the six categories used in this system and present in the Finnish Wikipedia. The topmost Wikidata identifier for each of these categories was manually identified, and they are used in SPARQL queries to the Wikidata Query Service. For example, the query below finds Wikidata entries that have a Finnish Wikipedia page and are classified as Person

⁴<https://sqlite.org/fts5.html>

⁵In its Finnish version, <https://fi.wikipedia.org/wiki/Wikipedia:Etusivu>

⁶https://www.wikidata.org/wiki/Wikidata:Main_Page

⁷<https://wikimediafoundation.org/>

⁸<https://dumps.wikimedia.org/>

("Q5"). (In practice, however, since the volume of responses is large, the query operation has to be carried out many times to avoid timeouts, using the LIMIT and OFFSET operations.)

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT DISTINCT ?item
WHERE {
  ?item wdt:P31 wd:Q5 .

  ?sitelink schema:about ?item;
            schema:isPartOf ?wiki .
  FILTER (?wiki IN (<https://fi.wikipedia.org/>)) .
}
```

This process is not perfect, since the open nature of Wikidata ends up leaving the scope of these categories to active users; as such, there are many instances of non-entities in the lists, such as "Kielikunta" ("Language family") or "Ohjelmointi" ("Programming"). In theory, however, these extra entries, even if selected as one of the candidates, should be disregarded by the LLM disambiguator. In this step, recall is more important than precision.

Once all entities from a certain category are identified, Wikipedia dumps are used to retrieve relevant information about each of them, including information on redirection and disambiguation pages: the former work as aliases for a certain entity, while the latter are hubs that list different entities under the same label. This information is used to populate a SQLite database, alongside identifiers and aliases from Wikidata and the entity categories explained above. This process is repeated for each of the six categories.

3.2.3. Wikidata database plugin

The Wikidata plugin works by gathering information from entities in Wikidata. The first step in its creation is the identification of entities in Wikidata, in a similar way to the Wikipedia plugin above, but recursively, including its sub-categories, and disregarding Wikipedia pages:

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?item
WHERE {
  { ?item wdt:P279* wd:{cat_id}}
}
```

A total of around 2,5 million entries was found via this process in January 2025. The number, types and specific entries will differ depending on the period the search is carried out.

The plugin gathers, for each entity, similar kind of information as the Wikipedia plugin: labels, categories, aliases, and descriptions in the languages chosen. The default is Finnish and English, with the latter working as backup.

3.2.4. Wikidata API plugin

This plugin works as a fallback for the previous plugins. It does not store any values locally: instead, it uses the Wikidata API to search for entities by label. It returns aliases and short descriptions of the candidates.

3.2.5. YAML

This is an ongoing part of the project, and has not yet been evaluated. In order to add useful information to the candidates, the predicates in the ontologies can be transformed into YAML-formatted text and added to the LLM prompt. We are using only first-level predicates, which mean that the predicate of a predicate is not taken into account. For example, if a country is identified as candidate, its heads of state will typically be included, among other pieces of information, as long as they appear in the ontology

at hand, but the parties to which they belong or their inauguration dates will not. This is an attempt at striking a balance between token load and information relevance.

We are interested in analyzing if the additional information load and the format they are presented in result in better accuracy than a paragraph of descriptive text. Furthermore, this would allow for the inclusion of ontologies that do not store textual descriptions, by simply including their statements about different entities.

3.3. Disambiguation and Linking

This step of the process starts with disambiguation, in order to find out which of the candidates, if any, corresponds to the entity. In this step, the labels and descriptions of the candidates are fed to an LLM, alongside dummy candidates that represent "None of the above" and "The entity in question is not a named entity". The LLM is instructed to explain its choice step-by-step, since this method has been shown to improve results (Wei et al. 2023). The explanation is to be based on the facts presented and avoid vague language such as "maybe" and "could". The LLM is also instructed to return its response in a specific format. The translated version of the LLM prompt template is reproduced in Appendix A.

This disambiguation process resembles RAG (Lewis et al. 2020) in giving the LLM information it needs for answering the prompt. Unlike RAG, it does not add entire chunks of text to the prompt, instead transforming the task into a multiple-choice question. RAG assumes that the answer sought is in a chunk of text or a combination of them; FiNEL asks a more pointed question, and expects a single entry as response. Both rely heavily on the ability of the LLM to reason about the information given.

3.4. Manual Review and Editing

The purpose of our research is to create not only a strong baseline for automatic annotation of Finnish documents but also a user interface that supports its use in applications where correctness of the results can be controlled by a human user. As such, the entity linking tool is integrated into a front-end interface that could enable users to seamlessly revise and correct the named entity links found in a document and then save the metadata for further use in DH research and applications. Such a tool can be used for example in cases where annotations are critically important, such as in dealing with legal documents or parliamentary data⁹ (Hyvönen et al. 2024). The user would be able to upload the text to the tool, review and edit entity links proposed by the tool, and then save the corrected metadata in a fashion similar to the pseudonymization tool ANOPPI (Oksanen et al. 2019), previously created by our research group.

Figure 2 illustrates the structure of the editing interface. Annotations are embedded into the text as clickable components that bring up a menu for different editing functionalities. The color of the annotations depends on the type of the group to which it is attached. Hovering over an annotation or the group it belongs to will highlight all the annotations contained in the same group.

Singular annotations can also be modified. In addition to deleting any unwanted annotations, the user can also modify the start and end points of the annotation as shown in Figure 3 if, for example, the entity identification step has either missed or included excess tokens for the entity. Annotations can also be migrated from one group to another (see Figure 4) if a named entity has been mistakenly linked to different candidates between annotations. It is also possible to unlink an annotation from a group (the next button to the right in the menu in Figure 4), forming a new group on its own, if two different entities have mistakenly been linked to the same candidate.

It is also possible to add completely new annotations in the UI, as illustrated in Figure 5. New annotations can be linked to any existing group already used in the text or a completely new group can be created by filling its information. The interface will also be able to call the backend so that new candidates can be generated for both existing and new entities. The text with its annotation components is then regenerated to instantly reflect this new annotation and possible new group.

⁹Our group has released, among others, the LawSampo (<https://lakisampo.fi/>) and ParliamentSampo (<https://parlamenttisampo.fi/>) data services and semantic portals which use linked data.

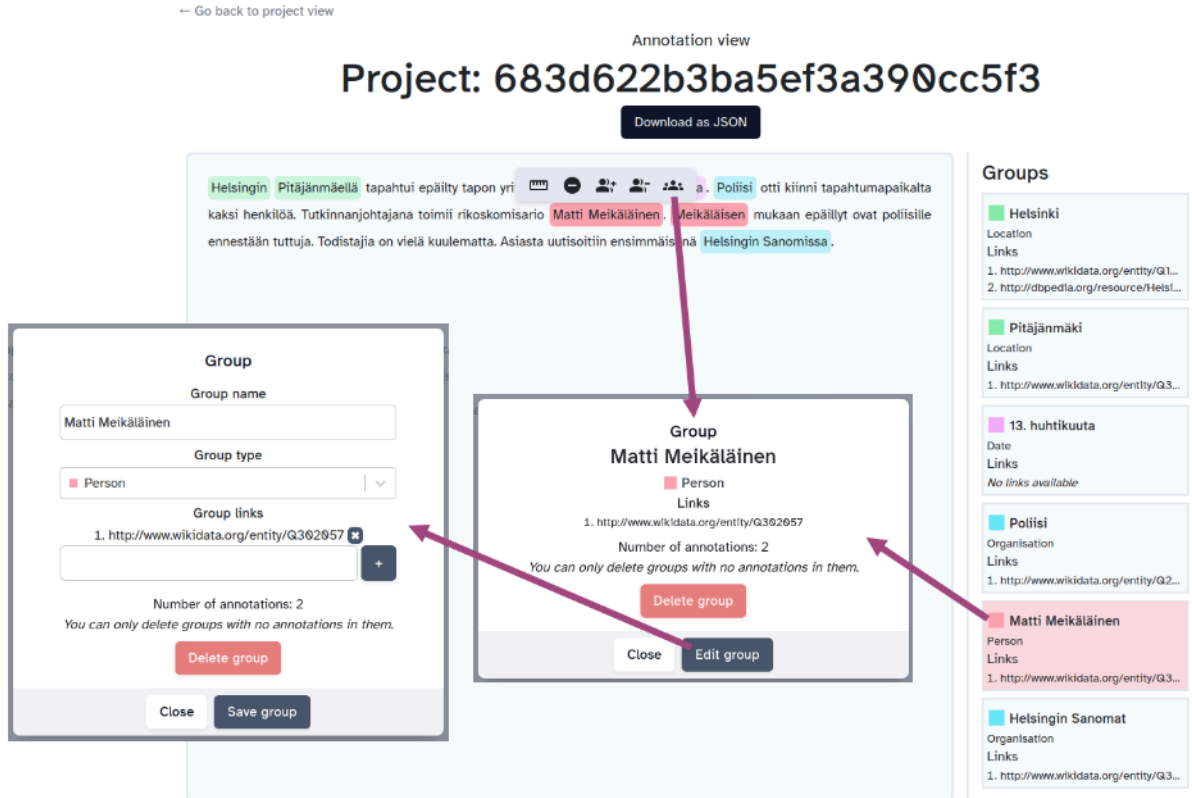


Figure 2: In the UI annotations are grouped based on the referenced entity and allows these groups to be edited.

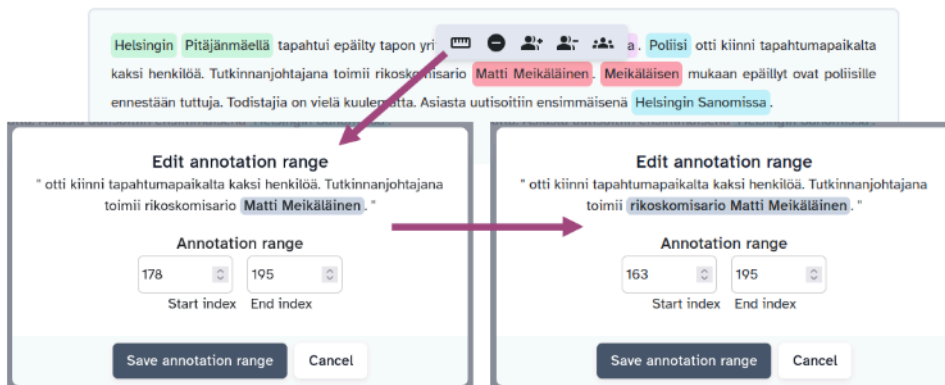


Figure 3: If an annotation is missing a part of the named entity (e.g., a word in front that should be part of its label), the annotation start and end points can be edited in the UI.

At the moment of writing, the finished annotated text can be downloaded in JSON format from the top of the page. New texts can be added by uploading pre-annotated JSON files in the same format that the UI outputs or by pasting and submitting plain text inside a dedicated tab for generating new text projects.

4. Evaluation

In this section, we evaluate two aspects of FINEL: Named Entity Lemmatization, which is essential for candidate generation, and Entity Disambiguation. The task of Named Entity Recognition has not been evaluated, since it uses a third-party tool that has undergone its own evaluation in Luoma et al. (2020). The results presented here are preliminary and may change as the system matures.

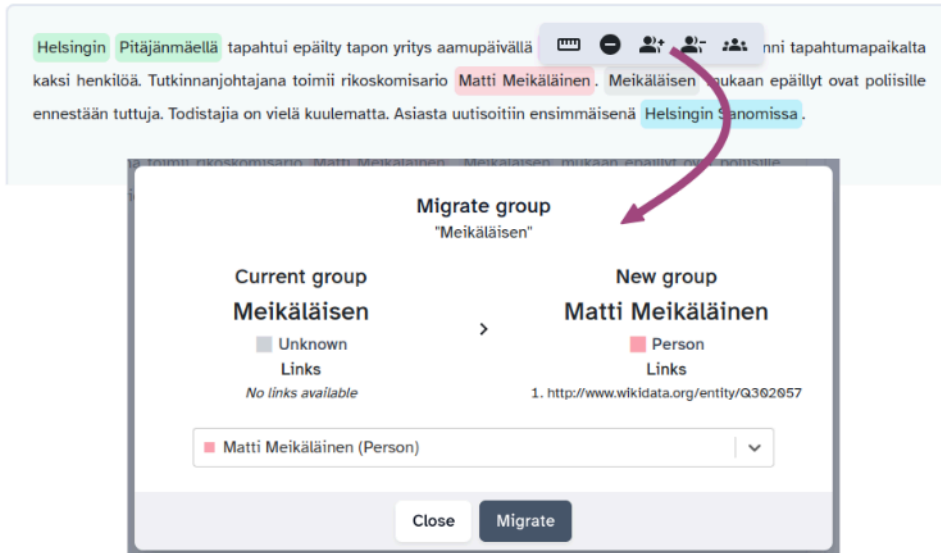


Figure 4: Annotations can be migrated to an existing group in the UI.

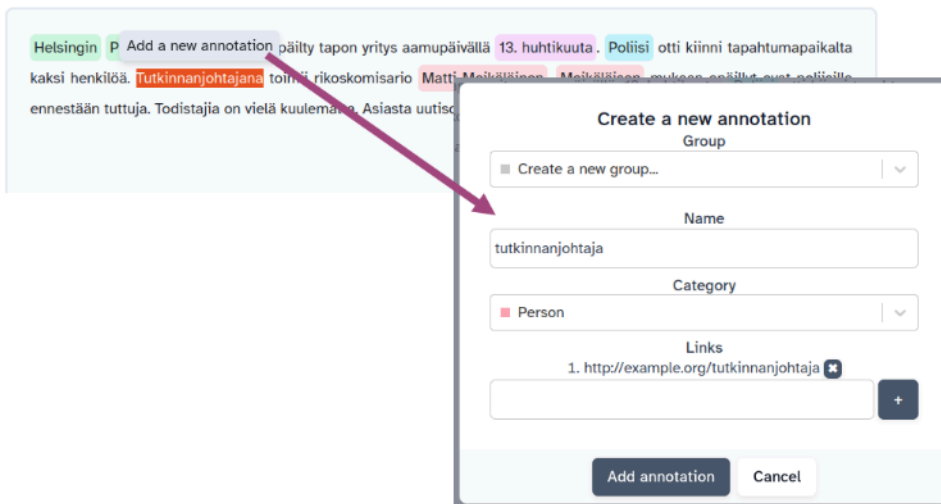


Figure 5: New annotations can be added using the UI, either linking new annotations to existing groups or creating new ones.

4.1. Named Entity Lemmatization

The evaluation of Named Entity Lemmatization was carried out using a test set of 10K Wikipedia internal links previously analogous to the training set. It obtained an accuracy of 96.5%. A qualitative analysis showed that most errors occur when handling non-Finnish names. Moreover, a similar test realized on a similarly fine-tuned multilingual T5¹⁰ with around 300K parameters, only obtained an accuracy of around 83%. This demonstrates that the knowledge of Finnish grammar incorporated in the models is essential for the success of this task.

4.2. Entity Disambiguation

The evaluation of the Entity Disambiguation step was carried out using a dataset of Featured Articles obtained in April 2024 from the Finnish Wikipedia, which was processed to recognize entities and generate candidates. The system was set up to reproduce the LLM linking task, presenting the human

¹⁰google/mt5-sma11 from <https://huggingface.co/google/mt5-small>

Table 1

Disambiguation evaluation with Llama3.3-70b, unique surface forms. N=349

# candidates	# entities	Accuracy (row)	Accuracy (cumul.)
1	274	0.97	0.89
2	31	0.77	0.59
3	15	0.73	0.47
4	9	0.56	0.33
5	5	0.8	0.24
6+	15	0.0	0.0

assessor with a random entity and its assigned candidates. "Not an entity" and "None of the above" were also given as alternatives. The assessor indicates which of the alternatives is correct and then compares it to the answers given by the LLM. Thus, this evaluation strictly does not assess recognition of entities, but only the linking task, which means that entities missed in the NER step were not manually annotated.

The model used in the evaluation is Llama3.3-70B, with 70 billion parameters. This model was chosen due to its open-weight nature, which means that it can be self-hosted instead of used exclusively via an API call, and its ratio of size-to-capabilities. The parameter count represents a good compromise between capability and computational demand, but very few open-weight models have been released with a similar number of parameters. Other models are still being evaluated.

Two evaluations were carried out, one that takes into account unique surface (i.e. inflected) forms of the entities (Table 1), and another with unique entities, regardless of their surface form (Table 2). These results show entities whose correct linking target was found among the candidates. Both tables have the following columns:

- **# candidates:** how many candidates were found for the entities.
- **# entities:** how many entities with these many candidates.
- **Accuracy (row):** the accuracy of this row (entities with this many candidates).
- **Accuracy (cumul.):** accuracy that takes into account entities in rows under this one. This is a bottom-to-top cumulative accuracy, weighted by number of entities.

The tables show a strong accuracy for entities with only one candidate, reaching 97%. This means that in these cases the LLM does not reject the candidate by choosing "None of the above", "Not an entity", or some other response as the answer. In both tables, however, there is a marked decay in the following rows, reaching 0% accuracy for entities with six or more candidates. This drop is due in part to the nature of the task, since introducing new entries in multiple choice questions naturally introduces more venues for error; but also to the reasoning capabilities of the model. This might be due, at least partially, to token overflow in the input of the LLM, which may happen more frequently as the prompt grows in size. We are investigating this possibility. In case the errors are due to intrinsic limitations in the reasoning capability of the model, a larger model would probably fare better. In any case, this decay is much more noticeable in Table 2, although the small quantities assessed make this table less reliable.

Table 1 shows the results for a total of 349 entity surface forms. In addition, 38 entities, or around 10% of the total, were identified to which the correct candidate was not found. This can be regarded as an evaluation of the candidate generation step, with the caveat that the entities found in Wikinews tend to be well known. In about 32% of these cases, the LLM chose a spurious candidate. These numbers are not included or reflected in the tables.

A more robust evaluation is underway, including a more thorough assessment of the types of answer given by the LLMs to further adjust the prompt.

In addition to evaluating FiNEL formally, the tool has been or is being applied to a number of applications, as discussed in the next sections.

Table 2
Disambiguation evaluation with Llama3.3-70b, unique entities. N=281

# candidates	# entities	Accuracy (row)	Accuracy (cumul.)
1	250	0.97	0.89
2	6	0.5	0.28
3	1	0.0	0.23
4	6	0.33	0.24
5	4	0.75	0.21
6+	14	0.0	0.0

The screenshot shows the LetterSampo portal interface. At the top, there is a search bar and navigation tabs for 'LETTERS', 'PEOPLE AND ORGANIZATIONS', 'FONDS AND COLLECTIONS', 'PLACES', 'DIGITAL EDITIONS', 'FEEDBACK', 'INFO', and 'INSTRUCTIONS'. The main header displays 'J. V. Snellman'. Below this, there are two facets on the left: 'Mentioned person (automatically)' and 'Mentioned place (automatically)'. The 'Mentioned person' facet is expanded, showing a list of names with checkboxes, where 'Georg Wilhelm Friedrich Hegel' is selected. The 'Mentioned place' facet is also expanded, showing 'Ruotsi' and 'Tukholma', with 'Tukholma' selected. The main content area displays a table of search results. The table has columns for 'Title', 'Sender', 'Recipient', 'Related entity', and 'Mentioned entity'. The results are filtered to show letters where the sender is Snellman, Johan Vilhelm, and the recipient is related to Hegel or Stockholm. The table also includes pagination controls and a 'TABLE' button.

Figure 6: Using LetterSampo portal for searching the letters of J. V. Snellman by faceted search. The facets are on the left and search results on the right, filtered by selecting Hegel as Mentioned person and Stockholm as Mentioned place

5. Applications: Enriching LetterSampo

To test the usability of FINEL in practical applications for DH research, it has been used to link named entities found in the letters of the critical edition of the works of the philosopher and statesman J. V. Snellman (1806–1881)¹¹, managed by the Snellman Institute. The resulting metadata was then incorporated to the in-use semantic portal *LetterSampo Finland – Finnish Nineteenth-Century Letters on the Semantic Web* (Hyvönen et al. 2025)¹².

In this application, named entities serve as facets in faceted search (Hearst et al. 2002), facilitating the filtering of letters based on their content—and, more specifically, the entities referenced within them. These entities also enable the linking of letters that mention the same individuals or locations, allowing for the visualization of letters on maps based on the coordinates of these linked places.

By establishing connections between entities, we can enrich their data with information from related sources, such as Wikidata for geographical coordinates. For instance, in Figure 6, the correspondences of J. V. Snellman are thematically grouped in the search results. This is achieved by selecting Georg Wilhelm Friedrich Hegel, the German philosopher who influenced Snellman, in the Mentioned person facet, and Stockholm (Tukholma in Finnish) in the Mentioned place facet.

The new facets derived from FINEL enhance the user experience on the portal, making it easier to

¹¹J. V. Snellman Kootut teokset: <https://snellman.kootutteokset.fi/>

¹²Portal available at: <https://kirjesampo.fi>

search for and navigate correspondences. Additionally, they assist experts in creating networks and graphs to analyze the texts more effectively.

6. Discussion

Our main objective in working on this project is to increase the digital presence of Finnish, making the existing metadata-processing tools for this language more robust and full-featured. However, working with a language which is so peripheral even in an European context is challenging, which can be seen in practice in the absence of tools and resources such as open datasets.

This paper describes an entity linking system in Finnish, which capitalizes on existing tools and models in order to extract named entities, generate candidates for them and choose the best candidate for linking. It showed that existing LLMs are capable of dealing with Finnish text satisfactorily. These results suggest the feasibility of the methods and tools presented here.

6.1. Future Plans

The main additions planned for the functioning of FiNEL: the comparison between different open-weights LLMs, to avoid committing to a single LLM family; the addition of predicates as YAML; the refinement of the input prompt, which would allow for both adding potentially important information about the candidates and utilizing knowledge graphs that do not store textual descriptions; and finally, our front-end interface for editing links will be finalized and integrated with our system. The system will be open-sourced in its entirety, alongside the datasets used to create it.

Additionally, another practical use case of FiNEL will be to enrich the textual speeches of the speakers at the Parliament of Finland in the already-existing PaliamentSampo. This will replace the current Named Entity Linking system, which recognizes Members of Parliament (MP) and places mentioned in the speeches. The results will be integrated with the analysis tools already existing in ParliamentSampo, including timelines, pie charts/histograms and maps, as customary in Sampo portals based on the Sampo-UI framework (Ikkala et al. 2022; Rantala et al. 2023) As can be seen in this case, additional metadata not only facilitates filtering speeches in useful ways as in the LetterSampo case mentioned earlier, but also makes it possible to study how the MPs and parties refer to each other in their speeches by using methods of Network Analysis (Poikkimäki et al. 2022).

Acknowledgments:

Our work is part of the national FIN-CLARIAH research infrastructure programme, funded by the Research Council of Finland. This project has received funding from the European Union – NextGenerationEU instrument and is funded by the Research Council of Finland under grant number 346323. CSC – IT Center for Science has provided computational resources for our projects. We also acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium through a EuroHPC Regular Access call.

References

- Ayoola, Tom, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. “ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking.” In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, 209–220. NAACL-HLT 2022. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/2022.naacl-industry.24>.

- Baek, Jinheon, Alham Aji, and Amir Saffari. 2023. “Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering.” In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, edited by Estevam Hruschka, Tom Mitchell, Sajjadur Rahman, Dunja Mladenić, and Marko Grobelnik, 70–98. MATCHING 2023. Toronto, ON, Canada: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/2023.matching-1.7>.
- Edge, Darren, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. “From Local to Global: A Graph RAG Approach to Query-Focused Summarization.” Pre-published, April 24, 2024. <https://doi.org/10.48550/arXiv.2404.16130>. arXiv: 2404.16130 [cs].
- Fan, Wenqi, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models.” In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6491–6501. KDD ’24. New York, NY, USA: Association for Computing Machinery, August 24, 2024. ISBN: 9798400704901. <https://doi.org/10.1145/3637528.3671470>.
- Hearst, M., A. Elliott, J. English, R. Sinha, K. Swearingen, and K.-P. Lee. 2002. “Finding the flow in web site search.” *CACM* 45 (9): 42–49.
- Hipp, D. Richard, D. Kennedy, and J. Mistachkin. 2024. *SQLite [Computer Software]*. V. 3.45.1, January 30, 2024. <https://www.sqlite.org>.
- Hyvönen, Eero. 2022. “Digital Humanities on the Semantic Web: Sampo Model and Portal Series.” *Semantic Web – Interoperability, Usability, Applicability* 14 (4): 729–744. <https://doi.org/10.3233/SW-190386>.
- Hyvönen, Eero, Petri Leskinen, Henna Poikkimäki, Heikki Rantala, Jouni Tuominen, Senka Drobac, Ossi Koho, Ilona Pikkanen, and Hanna-Leena Paloposki. 2025. “LetterSampo Finland (1809–1917) Data Service and Portal: Searching, Exploring, and Analyzing Historical Letters and Their Underlying Networks.” In *Proceedings of ESWC 2025, Supplement, Poster and Demo Papers*, Accepted, forthcoming. Springer-Verlag.
- Hyvönen, Eero, Laura Sinikallio, Petri Leskinen, Senka Drobac, Rafael Leal, Matti La Mela, Jouni Tuominen, Henna Poikkimäki, and Heikki Rantala. 2024. “Publishing and Using Parliamentary Linked Data on the Semantic Web: ParliamentSampo System for Parliament of Finland.” Accepted, *Semantic Web – Interoperability, Usability, Applicability*, <https://www.semantic-web-journal.net/system/files/swj3605.pdf>.
- Ikkala, Esko, Eero Hyvönen, Heikki Rantala, and Mikko Koho. 2022. “Sampo-UI: A Full Stack JavaScript Framework for Developing Semantic Portal User Interfaces.” *Semantic Web – Interoperability, Usability, Applicability* 13 (1): 69–84. <https://doi.org/10.3233/SW-210428>.
- Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. “Retrieval-augmented generation for knowledge-intensive NLP tasks.” In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS ’20. , Vancouver, BC, Canada, Curran Associates Inc. ISBN: 9781713829546.
- Logeswaran, Lajanugen, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. “Zero-Shot Entity Linking by Reading Entity Descriptions.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, edited by Anna Korhonen, David Traum, and Lluís Màrquez, 3449–3460. ACL 2019. Florence, Italy: Association for Computational Linguistics, July. <https://doi.org/10.18653/v1/P19-1335>.

- 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 12th Language Resources and Evaluation Conference*, 4615–4624. LREC 2020. Marseille, France: European Language Resources Association, May. ISBN: 979-10-95546-34-4, accessed November 30, 2021. <https://aclanthology.org/2020.lrec-1.567>.
- Mäkelä, Eetu. 2014. “Combining a REST Lexical Analysis Web Service with SPARQL for Mashup Semantic Annotation from Text.” In *The Semantic Web: ESWC 2014 Satellite Events*, edited by Valentina Presutti, Eva Blomqvist, Raphael Troncy, Harald Sack, Ioannis Papadakis, and Anna Tordai, 424–428. Cham: Springer International Publishing. ISBN: 978-3-319-11955-7. https://doi.org/10.1007/978-3-319-11955-7_60.
- Martinez-Rodriguez, Jose L., Aidan Hogan, and Ivan Lopez-Arevalo. 2020. “Information Extraction Meets the Semantic Web: A Survey.” *Semantic Web – Interoperability, Usability, Applicability* 11 (2): 255–335.
- Mihalcea, Rada, and Andras Csomai. 2007. “Wikify!: Linking Documents to Encyclopedic Knowledge.” In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, 233–242. CIKM07: Conference on Information and Knowledge Management. Lisbon Portugal: ACM, November 6, 2007. ISBN: 978-1-59593-803-9. <https://doi.org/10.1145/1321440.1321475>.
- Oksanen, Arttu, Minna Tamper, Jouni Tuominen, Aki Hietanen, and Eero Hyvönen. 2019. “Anoppi: A Pseudonymization Service for Finnish Court Documents.” In *Legal Knowledge and Information Systems. JURIX 2019: The Thirty-Second Annual Conference*, edited by M. Araszkievicz and V. Rodríguez-Doncel, 251–254. IOS Press, December. ISBN: 978-1-64368-048-4. <https://doi.org/10.3233/FAIA190335>.
- Poikkimäki, Henna, Petri Leskinen, Minna Tamper, and Eero Hyvönen. 2022. “Analyses of Networks of Politicians Based on Linked Data: Case ParliamentSampo – Parliament of Finland on the Semantic Web.” In *New Trends in Database and Information Systems*, edited by Silvia Chiusano, Tania Cerquitelli, Robert Wrembel, Kjetil Nørvgå, Barbara Catania, Genoveva Vargas-Solar, and Ester Zumpano, 585–592. Cham: Springer International Publishing. ISBN: 978-3-031-15743-1.
- Rantala, Heikki, Annastiina Ahola, Esko Ikkala, and Eero Hyvönen. 2023. “How to create easily a data analytic semantic portal on top of a SPARQL endpoint: introducing the configurable Sampo-UI framework.” In *VOILA! 2023 Visualization and Interaction for Ontologies, Linked Data and Knowledge Graphs 2023*. CEUR Workshop Proceedings, Vol. 3508. <https://ceur-ws.org/Vol-3508/paper3.pdf>.
- Ruokolainen, Teemu, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2020. “A Finnish News Corpus for Named Entity Recognition.” *Language Resources and Evaluation* 54, no. 1 (March 1, 2020): 247–272. ISSN: 1574-0218. <https://doi.org/10.1007/s10579-019-09471-7>.
- Tamper, Minna, Rafael Leal, Laura Sinikallio, Petri Leskinen, Jouni Tuominen, and Eero Hyvönen. 2022. “Extracting Knowledge from Parliamentary Debates for Studying Political Culture and Language.” In *Proceedings of the 1st International Workshop on Knowledge Graph Generation From Text and the 1st International Workshop on Modular Knowledge co-located with 19th Extended Semantic Conference (ESWC 2022)*, edited by Sanju Tiwari, Nandana Mihindukulasooriya, Francesco Osborne, Dimitris Kontokostas, Jennifer D’Souza, and Mayank Kejriwal, 3184:70–79. International Workshop on Knowledge Graph Generation from Text (TEXT2KG 2022). CEUR WS. http://ceur-ws.org/Vol-3184/TEXT2KG_Paper_5.pdf.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." Pre-published, January 10, 2023. <https://doi.org/10.48550/arXiv.2201.11903>. arXiv: 2201.11903 [cs].

A. Appendix: LLM Prompts

The basic disambiguation prompt is a Python Template defined as follows. This is a translation of the Finnish original done by Google Gemini 2.0 Flash:

```
instructions = Template(
    """
    You are proficient in the Finnish language and an expert in entity disambiguation.

    Given text: "$paragraph"

    Entity mentioned in the text: "$entity_form"

    Candidates for this entity:
    $candidates

    Choose the best candidate from the candidates above based on the information in the text.
    Justify your choice step by step and clearly.
    Write your answer in the following format:

    # Explanation: <Explanation of the answer, with justifications step by step and clearly.>
    # Answer: <Candidate number> <Candidate name>

    Note the following in your answer:

    * Use the exact number and name of the candidate in your answer.
    * Do not add extra characters or text to the answer.
    * Explain your choice in detail, referring to the information in the text.
      Why did you choose this particular candidate?
      Why are the other candidates not as good?
      What information in the text relates to this?
    * Avoid vague expressions such as "could be" or "possibly".
      Justify your choice factually.
    * If it is not a named entity, choose the candidate named
      "Kyseessä ei ole nimetty entiteetti" (This is not a named entity).
    * If none of the candidates match the entity, choose the candidate named
      "Ei mikään yllä olevista" (None of the above).
    $date

    Absolutely remember to format your answer according to the instructions.
    """
)
```

While the date section might exist or not depending on the information available:

```
date = Template(
    """
    * The date of this text is $doc_date (in YYYY-MM-DD or YYYYMMDD format).
    Exclude answers where a person's date of birth or an event's
    date is known and is later than $doc_date, unless the exception
    is clearly consistent.
    """
)
```