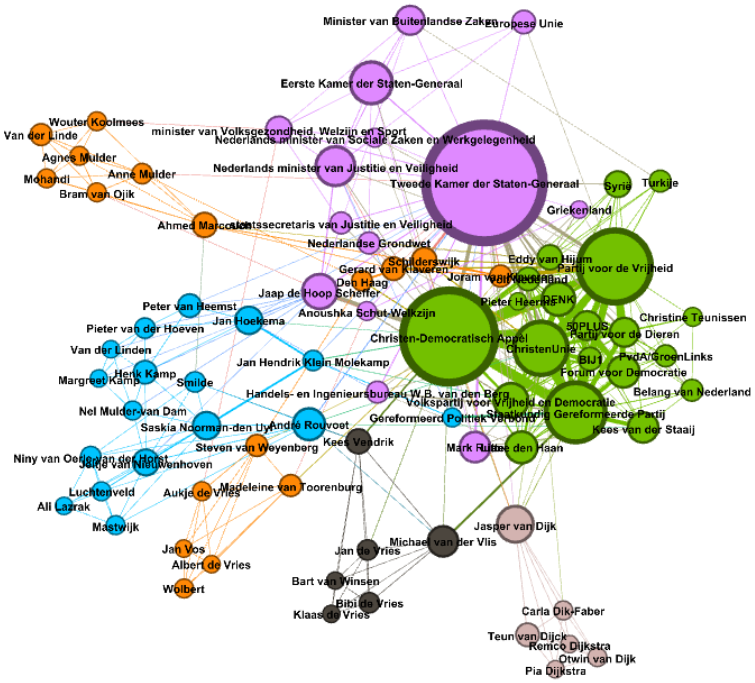
# Automatic co-occurrence generation using Named Entity Recognition and Entity Linking in Dutch Woo Documents

submitted in partial fulfillment for the degree of master of science

Daniël Drucker
13209221

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on 27.06.2025




| | UvA Supervisor |
|---|---|
| **Title, Name** | Maarten Marx |
| **Affiliation** | UvA Supervisor |
| **Email** | maartenmarx@uva.nl |

## ABSTRACT

Co-occurrence networks are used to represent (potential) relationships between terms, such as named entities, within a domain. Automatically generating such networks from text remains a challenging task. There already exists robust techniques for doing Named Entity Recognition (NER) and Entity Linking (EL) in text documents. This study proposes a full pipeline for recognizing persons, organization, locations, and laws in Woogle, a search engine for open Dutch government documents, to automatically generate co-occurrence networks for these entity types. The findings demonstrate that the developed pipeline can recognize entities with an f1-score of 0.7, while the EL component achieves a micro f1-score of 0.74, leaving further room for improvements. This paper also showed some evaluations of the co-occurrence networks that have been created.

## KEYWORDS

Named Entity Recognition, Entity Linking, Co-occurrence network, Woogle

## GITHUB REPOSITORY

https://github.com/daniel-5151/Woogle-NER-EL

## 1 INTRODUCTION

Woogle is a search engine for open government documents in The Netherlands under the Freedom of Information Act (Wet open overheid/Woo), which aims to improve the transparency of government actions and decisions. Like other search engines, such as Google, the user is able to type a search query and apply filters to retrieve the most relevant open government files in PDF format. Earlier research has shown that open government data has the ability to decrease corruption, increase transparency, and create better informed citizens [2].

These government files often contain important information such as names, organizations, and locations, known in the field of Natural Language Processing (NLP) as *Named Entities*. There are already robust techniques for recognizing these entities in documents, called Named Entity Recognition (NER) [18], and mapping variations in spelling and aliases to a single form, a task known as Entity Linking (EL) [25]. There has also already been research on co-occurrence networks, where entities that have a connection to each other are linked in a graphical network. However, these techniques have not yet been applied on documents retrieved from the Woogle search engine, and research on their use in other domains involving government files is also limited, leaving a research gap.

Solving this research gap can be implemented as an extension in Woogle, where named entities that often appear together in different documents or often appear close together in text can be viewed as a co-occurrence network in a user interface. A database of these entities can also be created, where each entity has a profile describing which documents it has produced and their ego-network, such as in Google Scholar.

This research will pose the following Research Question: *How can normalized entity information be used to construct co-occurrence networks and entity profiles in Dutch government documents?*

The research will be divided with the following sub-questions:

- **SRQ1:** How does statistical learning of Named Entity Recognition in Dutch Woo documents compare to Large Language Models in terms of performance and costs?
- **SRQ2:** To what extent can Dutch spelling variations be normalized/linked to a single form with a knowledge base?
- **SRQ3:** How can the extracted entities be integrated into a readable user-interface using co-occurrence and ego networks?

To answer these research questions, few-shot learning techniques will be employed for NER and EL using Wikidata as a Knowledge Base. The NER model will utilize a tok2vec model, while Dutch and multilingual BERT models are used for the task of EL. Furthermore, the resulting co-occurrence networks will be visualized and evaluated for different governmental dossier types.

## 2 RELATED WORK

### 2.1 Co-occurrence Network

The construction of co-occurrence networks has already been researched in scientific literature review with the use of keywords [24], where links are established when keywords appear together in an article. The number of times keywords co-occur in articles is also represented by weights. They also propose a chronological analysis of the network, making it possible to observe trends over time in a particular field.

Co-occurrence networks are also found in ecological literature, such as with networks of species [9] and microbial communities [3, 13]. However, these studies found that co-occurrence networks do not reproduce actual interaction networks and are thus insufficient for interpreting species interactions.

Another type of network is an ego network. These networks have already been used in social networks [1], which describe the social relationship between an ego (individual) and its alters (social connections). These networks are also often presented as graphs, where nodes represent the users, and an edge represent a social relationship. Ego networks are typically analyzed by three properties: the number of links the ego has, the strength of these links (weights) and the number of interconnections between alters [1]. Other work has also constructed a tool to make these ego-networks dynamic in time [27].

### 2.2 Named Entity Recognition in Documents

Named Entity Recognition (NER) is a task to recognize entities in text and also make a distinction between different types (eg. person, location, and organization). It is an often used pre-processing step for other downstream tasks [18]. There are already a number of methods used for NER in existing literature. Traditional methods for this task include rule-based (manually annotated), unsupervised learning, feature-based supervised learning, while more recent approaches focus on deep learning [18]. Some widely popular deep learning techniques in NER include Convolutional Neural Networks, Recurrent Neural Networks, Recursive Neural Networks

and Transformers. The advantage of using deep learning methods is that it is very effective at automatically learning named entities, without a lot of engineering skill and domain expertise, making it the current state-of-the-art in NER [18].

These deep learning techniques have already been implemented into NER in Dutch domain-specific text that this study will be compared to. A BERT-based (Bidirectional Encoder Representations) approach was used and fine-tuned to recognize entities in 18th century Dutch sailors text [12], which resulted in a model with an f1 score of 0.74. Other research by Provatorova et al. [23] also used fine-tuned BERT-based models for NER on historical Dutch text that achieved f1-scores between 0.7 and 0.85 based on the entity type and time period. In addition, other work [4] also used a pre-trained Dutch BERT model to fine-tune it with archaeological document annotations. Their model achieved an f1-score of 0.735.

A novel method to improve these domain-specific models is by using prompt engineering using Generative AI (GenAI) with few-shot learning, which has been applied in health records [14]. This method has already come close to the performance of deep learning approaches in the biomedical field, achieving a f1-score of .86 on medical problems, treatments and tests. The use of LLMs has also been researched with zero-shot, but the f1-scores still lack compared to the state-of-the-art deep learning methods [28].

### 2.3 Entity Normalization/Linking

Entity Normalization or Linking is a task in information retrieval that aims to link entity mentions to a single concept that is present in a Knowledge Base (KB). This KB can be an online encyclopedia, such as DBpedia (based on Wikipedia) or Wikidata [25]. Both DBpedia and Wikidata have lightweight Entity Linkers. Since Wikidata will be used as a KB in this research, OpenTapioc [6] will be used as a baseline model, achieving f1-scores of 0.4-0.5 on news datasets in English.

Similarly to NER, recent research has mostly focused on deep learning methods. These methods often follow a two-step approach: initially a number of potential candidates are retrieved from the KB to reduce the search space [25]. These candidates are then re-ranked to pick the most probable candidate. There are several lightweight techniques for choosing the candidates. The first of these is to use common Information Retrieval methods of fuzzy string matching [12], indexing the KB [15], or exact/partial matching [17]. A newer method is to use approximate nearest-neighbor search (ANN) proposed by Gillick et al. [10], where candidates are picked based on their proximity to the entity mention within the word embedding space using cosine similarity. Their method does have a high recall, but requires additional training data (the authors used all linked mentions of Wikipedia).

The step of entity re-ranking is to assign a score to each of the candidates using the mention and its context, where the candidate achieving the highest score is picked as the normalized entity. Older deep learning methods make use of Conventional and Convolutional Neural Networks [8, 17] , which show improvements over the older rule-based approaches, achieving accuracies between 80-90% on different biomedical datasets with non-exact matches. Other research on old Dutch sailors text by Hendrinks et al. [12] used active learning for the entity linking part, which achieved an f1-score of 0.846.

The current state-of-the-art approach in biomedical entity linking makes use of transformers, more specifically BERT. Fine-tuned BERT models in the biomedical field improved the accuracy 1-5% depending on different datasets and models [15], where the best scores were achieved using pre-trained models specifically for the biomedical field. Fine-tuning a BERT model in the biomedical field in Dutch was also researched by Hartendorp et al. [11], which used an automatically generated dataset for fine-tuning, achieving a classification accuracy of 55%. Other work [26] uses a cross-encoder for candidate selection, where the context of an entity mention and a description of the candidates are used to rank the candidates. This method achieved an accuracy score of 94.5%. A similar approach has been used for entity linking in multilingual newspapers and classical commentaries [16], but utilizing multiple Wikipedia sentences to compare to the context of the mention instead of a single-sentence description. However, the results of their approach (f1-scores between 0.4-0.6 depending on the language and dataset) still fall short of the performances seen in the biomedical field.

Similarly to NER, a novel approach is to use prompt engineering using GenAI [7]. The advantage of this method is that it uses the LLM built-in knowledge and thus does not need labeled training data. But the authors also investigated a few-shot approach, where GPT3.5 and Llama2-chat are finetuned using instructions of pre-existing annotated datasets, slightly improving it over the zero-shot approach.

Research on EL often also includes NER as a prerequisite step. However, much of the existing literature treats NER and EL as the only tasks, often ignoring downstream tasks outside of information retrieval. The performances of state-of-the-art techniques on Dutch text also still lack behind those using English text. The most similar to this paper is the research of Provatorova et al. [22], which automatically built a social network of persons from historical documents, mostly in Dutch. However, this has not yet been done on Woo/governmental documents, which often have, in addition to persons, other important types, such as organizations, laws, and locations. This gap in research on this domain presents further research into combining NER/EL to make co-occurrence networks in a single pipeline.

## 3 METHODOLOGY

The main contribution of this research is the construction of a pipeline to automatically generate co-occurrence networks in the Dutch governmental domain for persons (politicians more specifically), organizations, locations, and laws. The methods discussed in this section consist of three main parts to combine the techniques used that have been discussed in the related work section into a single pipeline. Firstly, NER is applied on the Woogle text data with the goal of returning a list per document with its unique entities. The second task is to normalize these entities to a single form. The last step is to make a view of the "actors/main players" in the Woogle documents, where single-form entities that share a number of documents or are within each others context are linked together. A schematic overview of the pipeline with an example can be seen in Figure 3.

| Type | # Pages | # Documents | # Dossiers | # Tokens |
|---|---|---|---|---|
| 2b | 319,141 | 53,000 | 52,107 | 319,663,415 |
| 2c | 150,548 | 2,719 | 1,511 | 52,507,377 |
| 2e-b | 44,126 | 12,944 | 12,830 | 14,358,957 |
| 2i | 1,123,490 | 44,068 | 7,636 | 304,219,632 |

**Table 1: Woogle dataset details.**

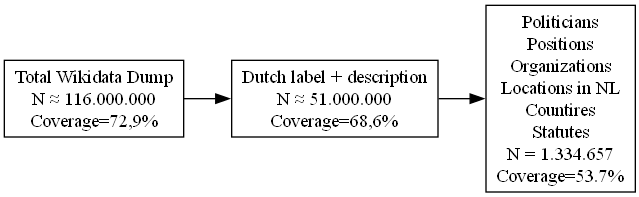



**Figure 1: Steps for filtering data in the knowledge base. The coverage indicates the percentage of entities in the test set that are present in the knowledge base and N shows the amount of items.**

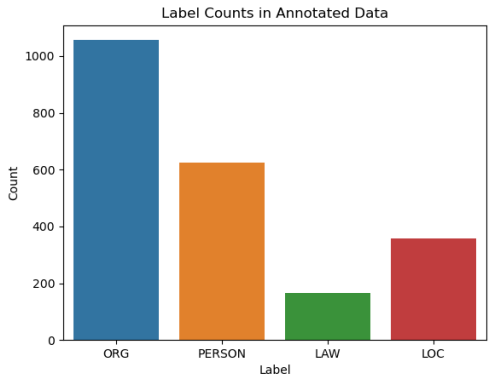



**Figure 2: The distribution of annotated entities in the training/validation set.**

## 3.1 Datasets

*3.1.1 Woogle Dump.* The source of the data is the Woogle collection, which contains text and metadata from Woo documents in csv file format [19]. This data goes up to March 12th 2025 as of the writing of this research and has multiple dossier types. This research will focus on the categories 2b, 2c, 2e-b,and 2i. These categories consist of meeting notes of the States General (First and Second Chamber of the Dutch parliament), local governments [1], government advisories, and Woo/Wob-requests respectively. Table 1 shows some details about each of these datasets.

Each page in the datasets contains a column with the underlying text of that page. This research will utilize the text that has been read from Optical Character Recognition (OCR), which is used to extract text from images of scanned PDF's [19]. Depending on the dossier type, there is a lot of new line (\n) and \uFFFF tokens. Because these tokens can be a hindrance for NER and text annotation, they are removes and replaced by a whitespace. Pages/rows that have a missing value for the OCR text are also removed, which accounts for about 7.5% of the total number of pages. In addition, pages of documents are joined, so that each row is now a document with its full text.

*3.1.2 Wikidata Knowledge Base.* The knowledge base for the task of entity linking will use the open-source data of Wikidata, which was chosen for the huge amount of entities available. The latest Wikidata dump has almost 116 million items as of the time of writing. In comparison, Wikipedia has 7 million items. This should in theory result in a higher recall. However, the vast amount of items in Wikidata also contains a lot of irrelevant items/entities not of interest in this research, which can lower precision and increase lookup time. For this reason, two main filtering steps were applied. The first was to select items that have both a label and a description in Dutch. Since we are only focusing on named entities of specific types, the knowledge base is further filtered to only include entities that are instances of those types or their subclasses. Figure 1 shows a more detailed flowchart of these filtering steps. For each entity, the label (page title), description and aliases are saved.

## 3.2 Named Entity Recognition

*3.2.1 Data Annotation.* Since there is no labeled data available for entities in the Woo domain, training and validation data need to be manually labeled to fine-tune and validate existing NER models. There are already strict guidelines for annotating different types of entities, such as the guidelines used by MaiNLP [21], which will be used for this task. In addition to these guidelines, job functions such as ministers, state secretaries, and mayors will also be annotated as persons. The pages used for annotating are randomly sampled across the four datasets. Manually annotating these pages is an expensive task, and thus only few-shot learning will be used with a total of 158 pages across the four dossier types. In addition, a small holdout set of 40 pages is also annotated.

*3.2.2 Model.* The model used for the extraction of entities in the documents is Spacy's statistical model [2], which has the option for built-in deep learning for NLP tasks that have also been used in previous research [20]. As mentioned in Section 2, these models perform slightly worse than current state-of-the-art NER techniques. However, Spacy's statistical model can train and run more efficiently on CPU's and it's existing NER language models are easy to fine-tune with little data. The pre-existing model used for this task to fine-tune is the `nl_core_news_lg`, which is trained on Dutch news.

After the training is complete, which is discussed into more detail in Section 3.6, the best model from the cross-validation will be utilized to run Spacy over the remainder of the pages. After an entity mention has been recognized, the mention and sentence will be used as input for the EL step.

In addition to training and testing statistical models, the task of NER in this domain will also be tested on OpenAI's GPT4o. Using these models for over 1.5 million pages in a browser or app would be

[1] Only notes from the municipality of Nijmegen are included in the dump

[2] https://spacy.io/api/entityrecognizer

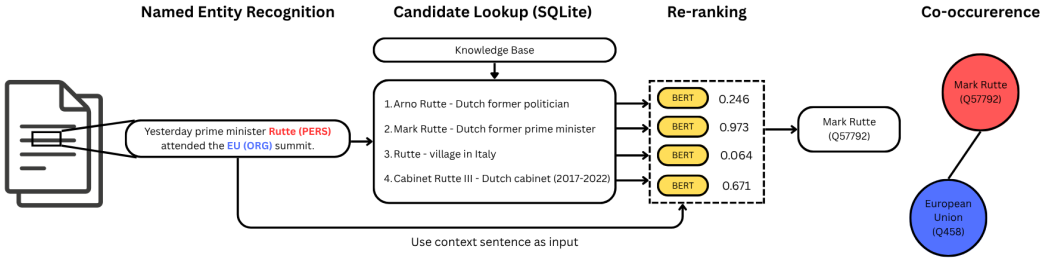



**Figure 3: Schematic overview of the full pipeline. A set of entities from the document are recognized, after which for each entity a number of candidates is generated. The descriptions and context are then used as input for a BERT model and a co-occurrence link is established between the normalized entities. In this example, the entity 'EU' also goes through the Candidate Lookup and Re-ranking steps to be normalized to 'European Union'.**

infeasible, and would thus need to be run with the API, which would come with financial costs. However, making an estimation of the performance and costs would give a comparison to cheaper methods and open the possibility for future research. The instruction/prompt given to GPT is to label the entities in a text sequence based on the annotation guidelines described earlier.

## 3.3 Entity Linking

As entity linking research is still somewhat fractured in the methods being used and variants in tasks, this implementation will mostly be a combination of previous work of this task.

*3.3.1 Candidate Lookup.* Since aliases for many entities are present in Dutch, we will use an alias table for the candidate selection. After the knowledge base is created, it is flattened so that each row is an alias as surface, also containing its associated QID, the original label, and description. Given the large amount of entities that have to be processed and the size of the knowledge base, the resulting dataset is stored in a SQLite dataset where the surface form is indexed.

The SQLite FTS5 extension is utilized to perform full-text search over the surface forms, using a recognized entity extracted from Spacy (the target surface). To ensure that each Wikidata item is only returned once (in the alias table it can occur that multiple instances of the same Wikidata page are returned if aliases of the same entity match the query), the results are grouped by the minimum score of their QID. The top $k$ pages with the lowest scores are returned and selected as candidate entities. The full query can be seen in Appendix B.

*3.3.2 Candidate Ranking.* In the second step, for each candidate a comparison is made between the context of the entity and the Wikidata descriptions of the candidates. This comparison is done by using fine-tuned cross-encoders. The purpose of these encoders is to check how similar the sentences are to see if both sentences refer to the same entity. The two models tested in this research are as follows:

- MS MARCO MiniLM L6 V2: This is a multilingual cross-encoder that has been trained on a large scale information retrieval corpus [3].
- BERTje: A Dutch BERT model that has been pre-trained on a large Dutch corpus [5].

Both models are fine-tuned by means of hard-negative mining. This data is obtained by sampling over random pages, extracting the entities using the fine-tuned NER model and obtaining suitable candidates described in the previous step. For each entity, the correct normalization is labeled as positive, while the other candidate descriptions are labeled as negative. If the correct entity is not in the top-$k$ candidates, all candidate descriptions are labeled as negative.

*3.3.3 Final Selection.* The final entity selection will be based on a linear combination of both the SQLite rank score and the similarity score of the cross-encoder. The final scores of the candidates will be calculated with the following formula:

$$score_{final} = \alpha * score_{encoder} + (1 - \alpha) * score_{FTS5} \quad (1)$$

In this formula, the encoder and FTS5 scores are normalized based on their respective minimum and maximum scores in the validation set. The value of $\alpha \in [0, 1]$ controls the relative contribution of the cross-encoder and FTS5. When $\alpha = 1$, the best candidate is selected based only on the cross-encoder scores. On the other hand, when $\alpha = 0$, the final score depends solely on the scores returned by FTS5.

In the final step, a threshold is applied to control if a candidate should be selected. Given a list of candidate entities $E$, the entity $\hat{e}$ with the highest score is only selected if the final score exceeds the threshold $t$. If none of the entities in the candidates list exceeds this threshold, the entity is discarded. The final selection is formally defined as:

$$\hat{e} = \begin{cases} \arg\max_{e \in E} score_{final}(e), & \text{if } \max_{e \in E} score_{final}(e) \geq t \\ \text{No match}, & \text{otherwise} \end{cases} \quad (2)$$

## 3.4 Co-occurrence Network

When the entities in a document have been normalized, the single form entities present in that document can be added as nodes in a co-occurrence network if that entity is not already present in the network. In addition to the single form, the locations of entities in

[3] https://www.sbert.net/docs/pretrained-models/ce-msmarco.html

a document are also saved. Two types of co-occurrence networks can be created for each dossier type:

(1) A document-level network in which an edge indicates entities that have appeared within the same document.
(2) A proximity-based network, where an edge represents entities have appeared together within $k$ words from each other.

This paper will mostly focus on the second type. The amount of times a co-occurrence takes place is indicated by the edge weight.

To make the network visualization look cleaner, all edges with a weight of 1 are removed. This significantly reduces noise and highlights stronger connections better. As a results, all nodes/entities that are only connected with low-weight edges are also removed. If there are still many more edges than nodes (more than 5:1), k-core decomposition is applied to remove low degree nodes. Finally, to make the graphs more readable, the Force Atlas 2 algorithm is applied. The size of individual nodes is also changed based on their degree, with high-degree nodes getting a larger size.

## 3.5 Evaluation

*3.5.1 Entity Extraction.* The NER model will be evaluated on recall, precision, and f1-score, since these metrics are widely used for these tasks and allow comparison of the performance to the current state of the art in the existing literature. The average performance and standard deviations of all folds of the validation sets are used to assess the overall performance of the model using 'exact match' evaluation (the boundaries and entity types must exactly match the ground truth). The small holdout set discussed in Section 3.2 will only be used for qualitative analyses of errors made by the model, since the size of this set could return biased metric scores. As for the Generative AI evaluation, the text of the test set will be prompted to the models and is compared to the ground truth of the annotations to also evaluate the metrics mentioned earlier.

*3.5.2 Linking.* The initial candidate lookup will be evaluated based on the recall of the golden-entity in the top-$k$ candidates of multiple limits. A validation/test set for this is created by extracting the predictions of NER on the holdout set discussed in the previous subsection and manually labeling the correct Wikidata page. The validation set consists of 303 examples, while the test set has 202 examples. The recall of the candidate retriever will be calculated only for entities that have an entry in the final knowledge base. The cross-encoders will be tested based on the Mean Rank and Mean Reciprocal Rank (MRR), where the rank of the golden-entity can also be evaluated. The final model will be evaluated based on micro f1-score, precision, and recall. In the context of this research, true positives, false positives, true negatives and false negatives are defined as in table 2.

| Gold | Prediction | Outcome |
|---|---|---|
| $Q_{xxx}$ | $Q_{xxx}$ | True Positive |
| $NILL$ | $NILL$ | True Positive |
| $Q_{xxx}$ | $Q_{yyy}$ | False Positive, False Negative |
| $Q_{xxx}$ | $NILL$ | False Negative |
| $NILL$ | $Q_{yyy}$ | False Positive |

**Table 2: Conditions under which outcome predictions fall.**

On the other hand, the f1-scores will be based on all classes. Since the task of EL has not yet been done on governmental texts and benchmark datasets do not exists for Dutch EL, we will compare the results on our own annotated test set[4] to that of previous work on newspaper/historical texts.

*3.5.3 Co-occurrence Network.* The co-occurrence networks are not evaluated with machine learning metrics, since there is also no ground-truth data available for this task. Instead, the network will be manually reviewed to check whether the network behaves as intended and no errors occur with edge cases. Since there are a lot of documents in the dataset the network will be expected to be very large, so this will be done on only a small subset of the data.

In addition we will look at the distribution of these metrics for each dossier type:

- Degree and Strength: The degree represents the number of links from a node and is used measures the importance of a node in a network. In a weighted network, the weight (strength) is also taken into account.
- (Local) Clustering Coefficient: This calculates how close the neighbors of a node are to forming a complete graph, giving an indication of the cohesiveness of a local neighborhood in the network.

## 3.6 Experimental Setup

This section will briefly go over some of the important parameters used during training for reproducibility. A more detailed list can be found in Appendix A.

**Training Splits:** Since the training dataset for the NER statistical model is quite small, 5-fold cross-validation will be applied on this task to provide a more robust estimate of the performance. At each fold, the data is divided into 80% for training and 20% for validation where every datapoint is used once in the validation set.

Since finetuning the cross-encoder is computationally more expensive and we do not get accuracy straight out of this encoder, finetuning will happen on just one fold, where the data is again divided in an 80/20 training and validation split.

**Training Setup:** Training of the NER model will be carried out using the annotated data mentioned in Section 3.2. The standard hyperparameters given by Spacy are used for training, with the exception of the batch size being reduced to 16 and the maximum epochs set to 10 as to prevent overfitting and speed up training.

The finetuning of the cross-encoders will be done by only running a maximum of 4 epochs and with a batch size of 16. The loss function that will be used is the `BinaryCrossEntropyLoss`, which is built-in the CrossEncoder package. The evaluation strategy is done with the `CrossEncoderCorrelationEvaluator`, also from the same package.

**Hyperparameter Tuning:** In the entity linking pipeline, there are three hyperparameters that need to be tuned: the query limit, $\alpha$ and the threshold $t$. These are selected by testing the effect of different values of these hyperparameters on the validation set accuracies. The exception here is the threshold, which will be evaluated based on the precision, recall and f1-score.

[4]The gold-link of mentions that have a Wikidata entity, but not an entry in the knowledge base are changed to NILL.

**Graph Visualization:** The Force Atlas 2 algorithm is run in *Gephi* using a scaling of 100 and a gravity of 150 to prevent islands from floating away. The nodes are scaled by their degree, with the lowest degrees having a size of 10 and the highest degree nodes having a size of 100. In addition, nodes are also colored based on their community using the modularity algorithm.

# 4 RESULTS

## 4.1 Named Entity Recognition

The overall results of the NER experiments are presented in Table 3. The fine-tuned model significantly outperforms the baseline model (`nl_core_news_lg`) by almost 30%. Interestingly, the zero-shot GPT-4 model performs on par with the fine-tuned statistical model. This is mostly due to a higher precision, although the recall is worse. Even though GPT-4 shows competitive results with no training data, scaling with its API requires substantial financial costs. For example, dossier type 2b alone has over 300 million tokens, making the estimated input cost over $600, although this could be reduced by implementing a cheaper model, like GPT-4 nano. This makes large-scale deployment on all documents financially expensive. For this reason, we will use the fine-tuned statistical model for further research.

| Model | F1-score | Precision | Recall |
|---|---|---|---|
| Provatorova et al. [23] | 0.73-0.90 | 0.76-0.94 | 0.71-0.90 |
| Brandsen et al. [4] | 0.735 | 0.743 | 0.729 |
| **Spacy Base** | 0.43 | 0.46 | 0.41 |
| **Finetuned** | 0.70 (±0.037) | 0.73 (±0.050) | 0.67 (±0.050) |
| **GPT-4** | 0.69 | 0.76 | 0.63 |

**Table 3: Overall NER performance metrics with a 95% CI for the finetuned model using the standard deviation of the folds compared to other Dutch domain specific research.**

From performing a qualitative analyses on the holdout set, four common classes of mistakes are found with the fine-tuned model:

- False Negatives: A gold annotation for which there is no overlapping predicted span. This the most frequent error, accounting for 47.7% of all errors. An interesting observation in the holdout set in that there is a wide variety in entities that are missed.
- False Positives: These are seen as predictions that do not have any overlap with a gold annotation. These seem to occur in text that have a word suggestive of an entity, but which are not entities within the context. For example, *Tijdelijke wet* ("Temporary law") is predicted as a law even though it is not an actual law. There are also a lot of abbreviations that are falsely seen as entities. An explanation for these errors might be that a large fraction of abbreviations in the training data are actually entities. False positives account for 28.8% of the total errors
- Boundary Errors: These errors negatively influence both the precision and recall. They occur when an entity has an overlapping span with a gold notation, but with the wrong start and/or end position. A mistake that often occurs is the addition of prepositions or articles (E.g., *Holland Festival* en het - *Holland Festival of the*), or other random words (*de Volkskrant van vanmorgen - The Volkskrant of this morning*). On the other hand, when the gold annotations are long, the predicted entities can leave out some of the words that are included in the golden annotation. About 26.6% of the total errors are due to this reason.
- Label Errors: The label that has been predicted does not match the label of the gold annotation. As with boundary errors, these influence both the precision and recall. The most common mix-ups are between persons and organizations and between laws and organizations. If the boundary is still correct, it has no influence on the rest of the pipeline. These are the least common errors, accounting for 8.5% of all errors [5].

| Entity Type | F1-score | Precision | Recall |
|---|---|---|---|
| **LOC** | 0.70 (±0.076) | 0.69 (±0.093) | 0.72 (±0.075) |
| **ORG** | 0.68 (±0.044) | 0.74 (±0.089) | 0.64 (±0.085) |
| **PER** | 0.81 (±0.048) | 0.83 (±0.074) | 0.79 (±0.058) |
| **LAW** | 0.28 (±0.102) | 0.39 (±0.158) | 0.23 (±0.099) |

**Table 4: Entity-wise NER performance for the finetuned model. Results include a 95% CI using the standard deviation of the folds.**

Table 4 shows the performance metrics per entity type. The fine-tuned models capture persons quite well, which is the best scoring entity type. It also has the lowest confidence intervals, meaning that the model's performance is the most consistent across the folds. The performance on organizations and locations seems to be consistent with the overall model's performance. The extraction of laws, on the other hand, performs by far the worst of the entity types for the finetuned model. The performance on this entity type also varies greatly, as can be seen from the confidence intervals. The best fold for example has an f1-score of 0.41 on laws, while the f1-score is 0.15 on the worst fold. An explanation for this can be the lack of laws in the annotated data, which can result in some of the folds having very little examples of laws to train on.

## 4.2 Entity Linking

Figure 4 shows the percentage of gold entities in the validation set that is present with different query limits, considering only those mentions that have a gold entity in the knowledge base. The results show that the recall increases with the query limit up to a top-$k$ of 10 before plateauing around 80-82%. What is also interesting is that in 53% of the cases, the best returned entity from FTS5 is the gold entity, as can be seen from the 53% recall for a query limit of 1. For queries where the entities were not retrieved, the cause is often one of the following:

- The spelling of certain entities does not match any of the aliases in the knowledge base. This includes, for example,

[5]The total percentage exceeds 100%, because single predicted entities can have both the wrong boundaries as well as the wrong label

abbreviations (eg.,'ministry of FIN') or long/complex names (eg., 'M.L.L.E. Veldhuijzen van Zanten-Hyllner').
- The fine-tuned Spacy NER model makes mistakes with the bounds of the entity, which makes it not match with any item in the knowledge base. Examples of this are where extra text is added or when two separate entities next to each other are seen as one entity (eg., 'Flevoland Friesland'). These boundary mistakes are discussed in more detail in Section 4.1. In theory these can be solved with applying fuzzy string matching, but this would be computationally a lot more expensive.
- Spelling mistakes or mistakes caused by OCR reading in the documents.
- Mentions with a lot of other potential matches (such as common last names).

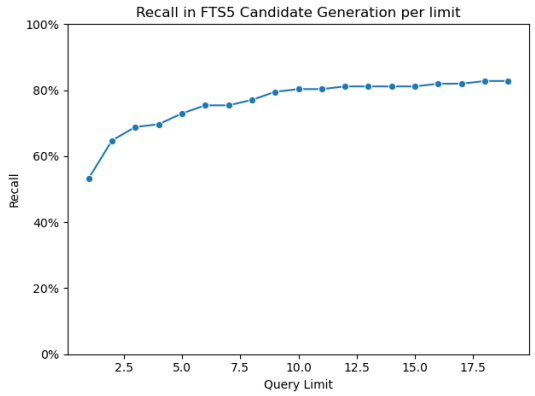



**Figure 4: Top-$k$ retrieval recall with FTS5 on the validation set.**

In Table 5, the performance of both transformer models that were tested are displayed. These performances are performed after the models are fine-tuned and with a maximum top-$k$ of 10, without any other hyperparameter fine-tuning (no threshold and an $\alpha$ of 1). In the table it can be observed that the Dutch BERTje transformer model outperforms the multilingual cross-encoder by a few percentage points. This better accuracy can be explained by also having a higher MRR and a lower Mean Rank, meaning that the gold entities are generally ranked higher in the final candidate ranking.

| Model | Accuracy | MRR | Mean Rank |
|---|---|---|---|
| MS-Marco-MiniLM | 0.67 (±0.053) | 0.66 | 1.35 |
| BERTje | 0.71 (±0.051) | 0.70 | 1.17 |

**Table 5: Performance of the pre-trained and finetuned transformer models, where the final score only uses the transformer scores ($\alpha = 1$). The accuracy uses a CI of 95%.**

The results for the hyperparameter $\alpha$ can be seen in Figure 5. An interesting observation is that both models do not appear to change in performance after a $\alpha$ value of 0.05 (the first value were the cross-encoder score are counts). When looking at the top candidate for both the FTS5 and cross-encoder scores, 42-49 percent are the same, depending on the model. However, the average Spearman correlation of the ranks for both models is around 0.13 to 0.18, suggesting that the cross-encoder still reorders a lot of the candidates.

The accuracy scores of the MS-Marco-MiniLM remains mostly stable at different values of $\alpha$, only fluctuating a few percentage points. The accuracy peaks between $\alpha$ values between $\alpha = 0.4$ and $\alpha = 0.6$, where the highest accuracy of 0.70 is observed at $\alpha = 0.55$. In contrast, the BERTje model remains flat until $\alpha = 0.75$ after which the performance significantly improves where the highest accuracy is 0.726 at $\alpha = 0.85$. However, when accounting for the confidence interval, there is no statistically significant difference between $\alpha = 1$ and other values of $\alpha$, other than 0. The significant lower accuracies of $\alpha = 0$ suggests that the re-ranking step is needed. For the rest of this paper, the values $\alpha$ with the highest 'mean' accuracies will be used for each model.

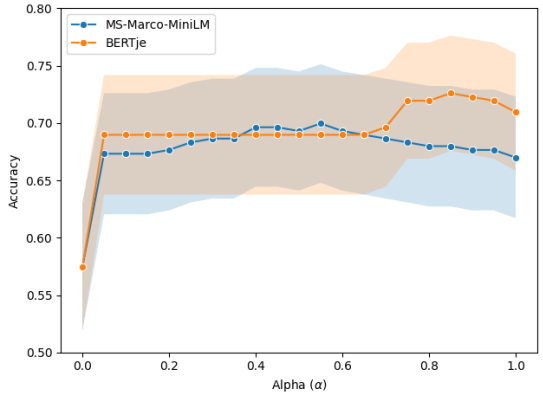



**Figure 5: Accuracy scores for both models across different values of $\alpha$ with a CI of 95% based on the size of the validation set.**

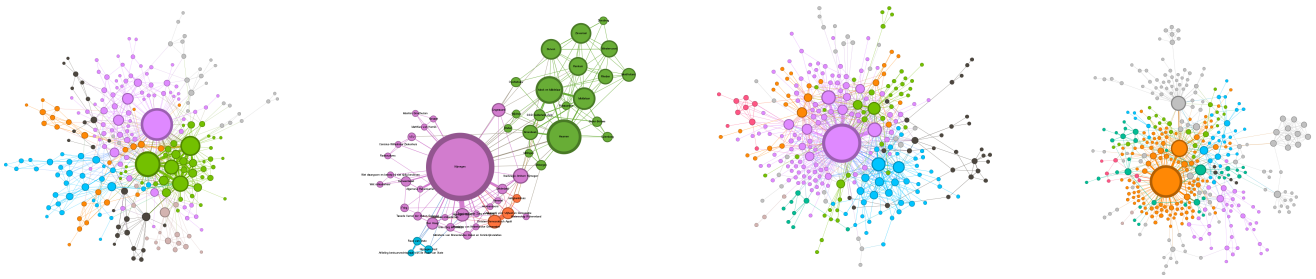
Figure 9 in Appendix C shows the distribution of scores of the correct and mislabeled prediction, both with the optimal $\alpha$ values. With both models, there is a slight difference in the scores, where mislabeled scores seem to overall rank lower than correct predictions. However, this difference is not very big and there is still a lot of overlap in the middle scores.

The precision, recall and f1-score across different thresholds are shown in Figure 10 in Appendix C. In the Marco-MS-MiniLM, although for the co-occurrence network the precision is more important, a threshold of 0.55 is selected as the optimal threshold. The same is true for the BERTje model, where recall also starts to drop significantly after a threshold of 0.6. This is chosen as the optimal threshold, since the precision still increases up to that point.

As can be seen in Table 6, the EL model using BERTje as a ranking adjuster performs slightly better than the MS-Marco-MiniLM model in terms of f1 score. With the optimal hyperparameters, MS-Marco-MiniLM has a better precision than BERTje, but also a worse recall. This means that the BERTje model misses a lot less relevant entities, but makes more mistakes when the threshold is met.

| Model | Micro f1-score | Precision | Recall |
|---|---|---|---|
| OpenTapioca (AIDA) [6] | 0.48 | - | - |
| VOC documents (Dutch) [12] | 0.846 | - | - |
| EntGPT-P (zero-shot) [7] | 0.819 | - | - |
| Woogle MS-Marco-MiniLM | 0.7 (±0.063) | 0.88 (±0.071) | 0.73 (±0.065) |
| Woogle BERTje | 0.74 (±0.065) | 0.82 (±0.068) | 0.82 (±0.068) |

**Table 6: Performance of the finetuned models (with optimal hyperparameters) compared to other work with a 95% CI. The models of this research have been tested on the manually annotated test set.**



**(a) Dossier type 2b** **(b) Dossier type 2c** **(c) Dossier type 2e-b** **(d) Dossier type 2i**

**Figure 6: Visualizations per dossier types of the main component, where the entities have been obtained on sampled documents. Dossier type *2b* was run on 120 sampled documents, *2c* was run on 500 sampled documents, and *2e-b* and *2i* were each run on 1,000 documents.**

## 4.3 Co-occurence Network

The visualizations of the sampled co-occurrence network for each dossier type are shown in Figure 6, where the entity linker was run using the optimal hyperparameters. High-quality and labeled networks can be found on the Github [6]. The visualizations and graphs is this subsection all have a proximity window where $k = 50$. Since during pre-processing the pages of individual documents were appended, it allows for boundaries to be detected across pages.

In each visualization, the islands that are not part of the main cluster have been removed. Notably, dossier types 2c, 2e-b and 2i each have one clear central node with a very high degree (*Nijmegen*, *House of Representatives*, and *The Hague*, respectively). On the other hand, dossier type 2b lacks a dominant central node.

In dossier type 2b (parliamentary debates), there is a community of nodes with a high degree that mostly contains political parties. These nodes also have high-weight edges between them, which can be expected in a parliamentary setting. There is also a community with government ministries, agencies, and ministers (both titles and names). Most outer nodes and communities are a mix of persons, locations, and a few organizations.

Although dossier type 2c (documents of the municipality of Nijmegen) is run over the same amount of documents as 2e-b, it has by far the fewest nodes. The green cluster on the top right of the network consists mainly of villages around the cities of Nijmegen and Arnhem. The cluster on the bottom left (purple, orange, and blue) is a combination of laws, government agencies, and locations in Nijmegen itself. Since the amount of nodes in the network before filtering on weights is around the same as in the other dossier type, it suggests that there are a lot of weak connections.

Dossier type 2e-b (government advisories) is in some ways similar to that of dossier type 2b, where political parties are mostly clustered together. Government ministries, agencies, and ministers do form communities, but these are, unlike dossier type 2b, more spread throughout the network. For example, some of the different ministers are on opposite sides of the network. In addition, this dossier type does have a clear central node with a high degree, while this is lacking is dossier type 2b.

Dossier type 2i (Woo/Wob-requests) does seem to have the most communities outside of the main cluster, which is expected since it has a large variety of different topics. The communities inside the network also have a bigger variety of entity types than in other dossier types.

According to Figure 7, dossier type 2b has the largest fraction of nodes that have a high strength/weighted degree. There is especially a peak at a weighted degree of 10 compared to the other two dossier types. Dossier types 2e-b and 2i, on the other hand, seem to almost follow the same distribution, with the only difference being that 2i has two entities with a higher weighted degree. Another difference of 2b is that it has an overall higher local clustering coefficient than the other dossier types, which can be seen in Figure 8. This is confirmed with the average, which is 0.45 compared to 0.28 and 0.27 of 2e-b and 2i respectively. This means that entities in 2b tend to cluster more together. A likely explanation for this is that parliamentary debates often contain enumerations of political

[6] https://github.com/daniel-5151/Woogle-NER-EL/tree/main/network-images

parties and politicians during voting processes, which was also observed during the annotation process.

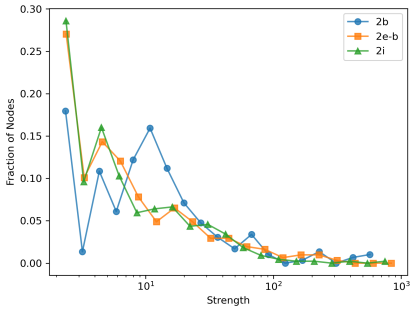



**Figure 7: Distribution of node strength for each dossier type network. Because of the low number of nodes, dossier type 2c is left out of this comparison.**

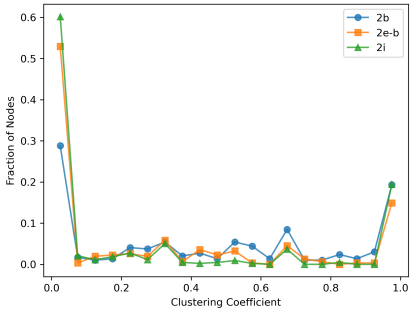



**Figure 8: Distribution of local clustering coefficient for each dossier type (excluding 2c).**

## 5 DISCUSSION

This study explored the NLP techniques of NER and EL to create co-occurrence networks on Dutch governmental documents in a few-shot setting. The domain-specific nature of this research makes it difficult to compare the findings of the NER component to related work. The results of NER indicate that it performs slightly worse than other Dutch domain-specific research, especially with the recall lacking behind. One reason for this could be the methods used, as the other Dutch models used more state-of-the-art BERT models, while this study opted for a more computationally efficient word2vec model. The inclusion of laws, which is the worst scoring entity type, might have also hurt the overall performance.

It should be noted that comparisons in EL are also difficult, since there is still a wide variety of task variants [10]. Compared to other similar tasks in multiple languages, the model of this study falls somewhere in the middle in terms of performance. The model performs better than the baseline OpenTapioca [6], although it should be noted that our knowledge base is substantially smaller. This is also the case with CLEF HIPE [16], where our method achieves a higher performance. However, this might also be explained by the difference in tasks, since their method is multilingual and has a substantially higher entity coverage in the knowledge base. The results of our EL methods fall short on other work on Dutch historical VOC text [12] by about 10 percentage points. This difference may again be explained by knowledge base size: Their approach uses a smaller knowledge base that is reduced even further with preliminary filtering per entity mention. Our method also has a worse performance than English EL using GenAI [7], whilst also needing more training data, as well as the current state-of-the-art deep-learning methods in the English biomedical field.

### 5.1 Limitations

The need to manually annotate data for NER and EL is a time-intensive task. Due to time constraints, both methods were evaluated on relatively small testing datasets. Moreover, the process of manual annotation can introduce errors in the datasets. Combining these challenges with the lack of external Dutch-language datasets for NER and EL poses a risk to the reliability of the results. This negative impact was somewhat mitigated by using cross-validation in the NER component and the use of confidence intervals in the EL pipeline. However, it remains unclear how the models perform outside the domain of Dutch governance. Using larger evaluation sets that also span outside the governmental domain can improve the reliability of the results in future work.

Moreover, the sampling methods used for training and evaluation may pose a threat to external validity. Although the training and validation test datasets were created with random sampling on different seeds, the method still poses the risk of introducing bias into these sets. Some entities, such as political parties, positions in government, and locations, occur frequently in Woo documents and may therefore be overrepresented in the training data. As a result, these entities are more likely to be correctly recognized and linked in comparison to less often occurring entities in the sampled documents. This can also have an effect on the co-occurrence network, where entities that are more likely to be correctly identified can have a higher (weighted) degree due to the fact of better recognition as opposed to semantic centrality.

Lastly, there are some ethical concerns with training the models in this research. Some dossier types, such as 2b and 2e-b mostly contain politicians and other public figures, for whom it is reasonable to assume their names can be used for training and fine-tuning deep learning models. However, dossier types 2c and 2i contain a lot of ordinary citizens and small businesses. Although the documents they appear in are public domain, the use of such names for training deep learning models without consent raises ethical and legal concerns. Therefore, these considerations should be taken into account when deploying these models on these dossier types, possibly requiring retraining.

## 6 CONCLUSION

This research aims to answer how entities in Woo documents can be normalized/linked to form co-occurrence networks. To answer SRQ1, a model for doing Named Entity Recognition had to be fine-tuned for this domain specifically using few-shot learning with manually annotated data. Although the performances of this model are a bit behind other work in Dutch specific domains, it is still effective and efficient at recognizing important entities within the

Woo documents. Using GenAI as an entity recognizer was also explored, where in a zero-shot setting GPT-4 was able to get a higher precision. However, scaling it further is more challenging due to pricing. An interesting avenue for future research would be to use GenAI for automatically annotating training data that is passed to a word2vec or BERT model. This method has the potential to significantly increase the training data without the need for labor intensive manual annotation.

Secondly, to address SRQ2, the use of Information Retrieval techniques was investigated to link the recognized entities to a Wikidata item/entity, also within a few-shot learning environment. For the selection of candidates, an indexed alias table was used primary because no additional training data was needed. The effectiveness of two fine-tuned BERT models was also demonstrated, with BERTje slightly performing better than the multilingual BERT. Both models were also given a prediction threshold, which slightly increased the precision for both models.

To answer SRQ3, we looked at how these recognized and normalized entities could be used to make profiles of each entitie. This was done by constructing co-occurrence networks on sampled documents from each of the relevant dossier types, where an edge represents a co-occurrence of two entities within 50 tokens. To remove noise and improve the visualizations, low-weight edges were removed.

Overall, Woogle documents lend themselves well to NER and EL in a few-shot learning setting to create co-occurrence networks of entities, but this still depends on the dossier type. Future work should focus on further improving the performances of both the entity recognizer and linker. Recall of the entity linker could be improved by further expanding the custom knowledge base. Since this could have a negative impact on precision, the cross-encoders could also be further fine-tuned by utilizing Wikipedia text as additional training data. Another avenue that could be explored with EL on these documents is the use of GenAI, which as benefit includes that no training data is required. However, this also introduces substantial financial costs with current OpenAI's API pricing. In addition, the performance of these models and structure of the co-occurrence networks could also be investigated on other dossier types to evaluate the generalizability.

## REFERENCES


[1] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. 2012. Analysis of ego network structure in online social networks. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE, 31–40.

[2] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. 2015. A systematic review of open government data initiatives. *Government information quarterly* 32, 4 (2015), 399–418.

[3] David Berry and Stefanie Widder. 2014. Deciphering microbial interactions and detecting keystone species with co-occurrence networks. *Frontiers in microbiology* 5 (2014), 219.

[4] Alex Brandsen, Suzan Verberne, Karsten Lambers, and Milco Wansleeben. 2022. Can BERT dig it? Named entity recognition for information retrieval in the archaeology domain. *Journal on Computing and Cultural Heritage (JOCCH)* 15, 3 (2022), 1–18.

[5] Wietse De Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582* (2019).

[6] Antonin Delpeuch. 2019. Opentapioca: Lightweight entity linking for wikidata. *arXiv preprint arXiv:1904.09131* (2019).

[7] Yifan Ding, Amrit Poudel, Qingkai Zeng, Tim Weninger, Balaji Veeramani, and Sanmitra Bhattacharya. 2024. Entgpt: Linking generative large language models with knowledge bases. *arXiv preprint arXiv:2402.06738* (2024).

[8] Arnaud Ferré, Louise Deléger, Robert Bossy, Pierre Zweigenbaum, and Claire Nédellec. 2020. C-Norm: a neural approach to few-shot entity normalization. *BMC bioinformatics* 21, Suppl 23 (2020), 579.

[9] Mara A Freilich, Evie Wieters, Bernardo R Broitman, Pablo A Marquet, and Sergio A Navarrete. 2018. Species co-occurrence networks: can they reveal trophic and non-trophic interactions in ecological communities? *Ecology* 99, 3 (2018), 690–699.

[10] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506* (2019).

[11] Fons Hartendorp, Tom Seinen, Erik van Mulligen, and Suzan Verberne. 2024. Biomedical Entity Linking for Dutch: Fine-tuning a Self-alignment BERT Model on an Automatically Generated Wikipedia Corpus. *arXiv preprint arXiv:2405.11941* (2024).

[12] Barry Hendriks, Paul Groth, Marieke van Erp, et al. 2020. Recognizing and Linking Entities in Old Dutch Text: A Case Study on VOC Notary Records.. In *COLCO*. 25–36.

[13] Hokuto Hirano and Kazuhiro Takemoto. 2019. Difficulty in inferring microbial community structure based on co-occurrence network approaches. *BMC bioinformatics* 20 (2019), 1–14.

[14] Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. 2024. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association* (2024), ocad259.

[15] Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. Bert-based ranking for biomedical entity normalization. *AMIA Summits on Translational Science Proceedings* 2020 (2020), 269.

[16] Kai Labusch and Clemens Neudecker. 2022. Entity Linking in Multilingual Newspapers and Classical Commentaries with BERT.. In *CLEF (Working Notes)*. 1079–1089.

[17] Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. 2017. CNN-based ranking for biomedical entity normalization. *BMC bioinformatics* 18 (2017), 79–86.

[18] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering* 34, 1 (2020), 50–70.

[19] M. Marx. 2023. Woogle dump. https://doi.org/10.17026/dans-zau-e3rk

[20] Eleni Partalidou, Eleftherios Spyromitros-Xioufis, Stavros Doropoulos, Stavros Vologiannidis, and Konstantinos Diamantaras. 2019. Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy. In *IEEE/WIC/ACM International Conference on Web Intelligence*. 337–341.

[21] Siyao Peng, Zihang Sun, Huangyan Shan, Marie Kolm, Verena Blaschke, Ekaterina Artemova, and Barbara Plank. 2024. MaiNLP NER Annotation Guidelines.

[22] Vera Provatorova, Carlotta Capurro, and Evangelos Kanoulas. 2024. The art of connections: constructing a social network from the correspondence archive of Sybren Valkema. *arXiv preprint arXiv:2410.13980* (2024).

[23] Vera Provatorova, Marieke Van Erp, and Evangelos Kanoulas. 2024. Too Young to NER: Improving Entity Recognition on Dutch Historical Documents. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA)@ LREC-COLING-2024*. 30–35.

[24] Srinivasan Radhakrishnan, Serkan Erbis, Jacqueline A Isaacs, and Sagar Kamarthi. 2017. Novel keyword co-occurrence network-based methods to foster systematic reviews of scientific literature. *PloS one* 12, 3 (2017), e0172778.

[25] Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web* 13, 3 (2022), 527–570.

[26] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814* (2019).

[27] Yanhong Wu, Naveen Pitipornvivat, Jian Zhao, Sixiao Yang, Guowei Huang, and Huamin Qu. 2015. egoslider: Visual analysis of egocentric network evolution. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 260–269.

[28] Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot NER with ChatGPT. *arXiv preprint arXiv:2310.10035* (2023).

## Appendix A TRAINING DETAILS

| Settings | Spacy NER model | Cross-encoders |
|---|---|---|
| Training batch size | 64 | 16 |
| Validation batch size | 64 | 32 |
| Maximum epochs | 10 | 4 |
| Learning rate | 1e-3 | NA |
| Dropout | 0.1 | NA |
| Evaluation frequency | 200 | Epoch |

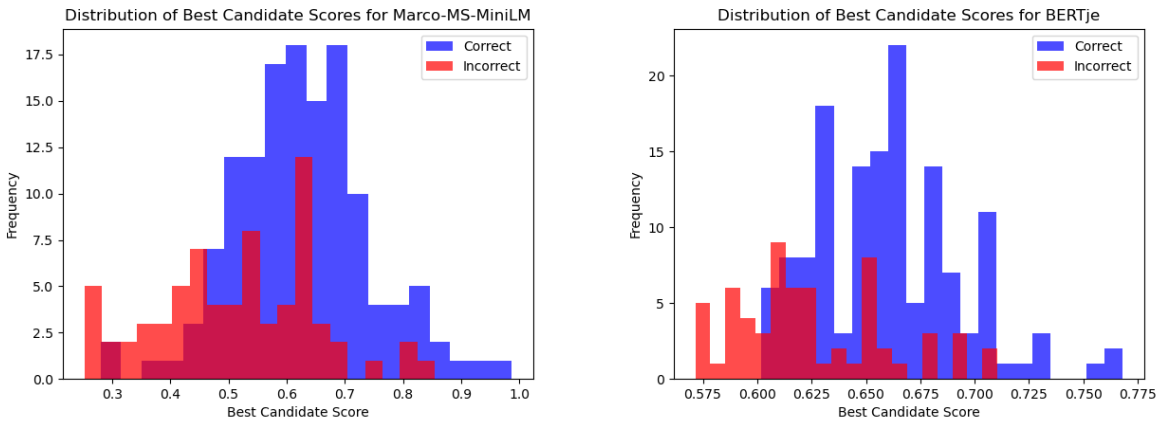
**Table 7: Main training hyperparameters for both NER and EL tasks. NA means that these hyperparameters were not explicitly initialized and thus use are the standard hyperparameters of the package/library.**

## Appendix B QUERIES

```sql
SELECT qid, MIN(rank) as best_rank, label, normal_form, description
    FROM kb
    WHERE surface_form MATCH ?
    GROUP BY qid
    ORDER BY best_rank
    LIMIT ?
```

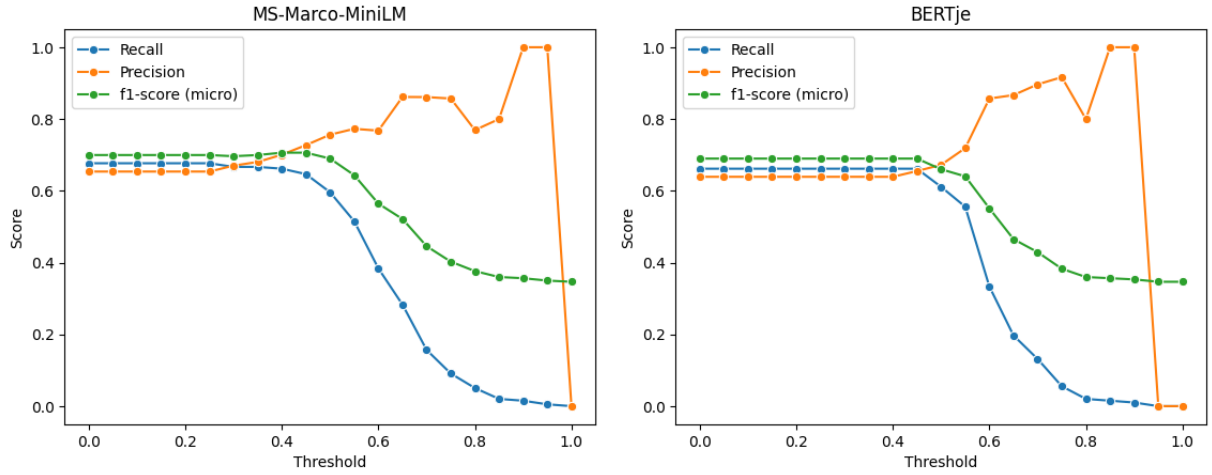
**Listing 1: SQL prompt used for retrieving possible candidates.**

## Appendix C FIGURES




**(a)** **(b)**

**Figure 9: Distribution of the best candidate score predictions divided in correct and incorrect in the validation set. With both models, there are clearly more incorrect predictions when the final score is at the low end. However, when the final score is somewhere in the middle there is still a lot of overlap between correct and incorrect predictions.**




(a) (b)

**Figure 10: Precision and recall for different thresholds on the validation set.**

## Appendix D REFLECTION ON THE USE OF GENERATIVE AI

In this research, the use of Generative AI was mainly used to improve productivity. More specifically, OpenAI's ChatGPT was used primarily for creating/improving visualization and debugging Python code. In addition, GenAI was also used to brainstorm ideas, but the final decisions on implementations were solely made by the author. GenAI was also not used in generating the text for the paper, only for checking spelling/typing errors (from which no text was copied and pasted in the report). This of course excludes the research of using GenAI models in the task of NER.