Geoparser: A Transformer-Based Bi-Encoder Approach for Efficient Toponym Disambiguation

GEO 511 Master's Thesis

Author
Diego Gomes
16-725-202

Supervised by
Prof. Dr. Ross Purves
Dr. Michele Volpi (michele.volpi@sdsc.ethz.ch)

Faculty representative
Prof. Dr. Ross Purves

30.09.2024
Department of Geography, University of Zurich

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Ross Purves, for his unwavering support and invaluable guidance throughout this thesis. His expertise and insightful feedback were instrumental in shaping this work, and his encouragement to explore new ideas helped broaden my perspective on the subject. I am also grateful to Dr. Michele Volpi, my co-supervisor, for his helpful input and advice during the course of this research.

A special thank you goes to my friends for their constant support and motivation. In particular, I want to acknowledge Nicolas Spring for his contributions as an early open-source collaborator on the software developed during this thesis, which enhanced the project in meaningful ways. I am also thankful to Aiyana Signer for the interesting discussions and stimulating conversations that helped me refine my ideas along the way.

Lastly, I would like to thank my family for their unconditional support, love, and encouragement, which sustained me throughout this journey.

Abstract

Information retrieval systems face significant challenges when interpreting toponyms in unstructured text due to their inherent ambiguity and context dependency. Traditional methods often struggle with the linguistic complexity involved, making it difficult to resolve these ambiguities effectively. While transformer models offer advanced linguistic capabilities, they are computationally expensive, and machine learning models, in general, often face difficulties in generalising across different domains. This thesis addresses these challenges by exploring how transformer models can be efficiently and effectively applied to toponym resolution and how their capabilities transfer across domains. A new method is proposed that uses a bi-encoder architecture within the SentenceTransformers framework to efficiently compare contextualised toponyms with potential location candidates from a gazetteer. This approach reduces computational demands by encoding toponyms and candidates separately, allowing for scalable similarity comparisons. The method is integrated into an end-to-end geoparsing pipeline through the development of the Geoparser Python library, which leverages spaCy for toponym recognition and provides functionalities for customisation and adaptation to specific text corpora. Experiments were conducted replicating a standardised evaluation framework to assess the performance of the proposed method. The results demonstrate that Geoparser achieves competitive performance compared to state-of-the-art systems, particularly excelling in computational efficiency. Further experiments show that the transformer model's performance can be enhanced for different domains with minimal additional training data. This work highlights the potential of transformer models for efficient and effective toponym resolution, offering promising directions for future research in Geographic Information Retrieval.

Table of Contents

1 Introduction

Geographic information is embedded in a wide variety of text documents, from news articles and posts on social media to historical documents. Finding and interpreting this information in unstructured text offers valuable insights for understanding and assessing the spatial relevance of documents for different information needs. Geographic Information Retrieval focuses on developing methods and techniques for extracting and using geographic information in text, for example, to improve web search engines (Purves et al., 2018). A key component of this process is geoparsing, which involves identifying and interpreting location references, primarily in the form of place names, often termed toponyms.

Toponyms are often ambiguous and can thus be used in different contexts to refer to different locations. Accurate resolution, therefore, requires a good understanding of the linguistic and geographical context in which they are used. However, traditional approaches to toponym resolution are often limited in this respect, especially when dealing with linguistically complex expressions (Gritta et al., 2018b). The advent of transformers in natural language processing has opened up new possibilities for addressing these challenges. Transformer models have stood out for their ability to generate contextualised representations of words and entire texts that capture complex syntactic and semantic structures (Devlin et al., 2019). This presents a promising potential for addressing the ambiguity of toponyms and thus enhancing the performance of toponym resolution systems.

Nonetheless, several key issues remain in employing transformers for toponym resolution. First, transformer models are often computationally expensive, limiting their practical application for large text corpora, especially in resource-constrained environments. Secondly, texts can vary in structure, theme, and geographical scope, which can be a substantial limitation for the generalisability of machine learning-based methods. Third, objective and reproducible evaluation of new techniques is crucial to assess their performance compared to existing systems; however, the use of different test datasets and metrics often makes it difficult to compare them effectively.

In view of this, the following research questions arise:

1. How can the linguistic capabilities of transformer models be used for toponym resolution in an efficient and still effective way?
2. How does the ability to resolve toponyms transfer to different text domains, and how can systems be adapted for this purpose with limited effort?
3. How can a new toponym resolution method be evaluated in a comparable way to allow an objective assessment of its performance compared to other systems?

This work seeks to explore these research questions. To this end, a new method for toponym resolution was developed with the aim of using transformer models in a computationally efficient way. This method was integrated into a complete end-to-end geoparsing pipeline in the form of the newly developed Geoparser Python library, which provides functionalities that enable users to adapt different pipeline components to the specific requirements of different text corpora. Finally, the method was evaluated by replicating the evaluation framework developed by Hu et al. (2023a) to compare the performance with those of state-of-the-art systems.

The thesis is structured as follows: Chapter 2 first provides some theoretical background, after which Chapter 3 gives an overview of the current research on transformer-based toponym resolution. Chapter 4 presents the proposed method and the new Geoparser library, followed by a description of the experiments to evaluate the method in Chapter 5. The results are presented in Chapter 6 and discussed in Chapter 7. Finally, the conclusions are drawn in Chapter 8.

2 Background

2.1 Geographic Information Retrieval

Vast amounts of information are freely and easily accessible online. Much of this information comes in the form of unstructured texts, such as news articles, blog posts or scientific publications. For information retrieval (IR) systems, content of this kind presents a significant challenge. The lack of a schematic data format means that unstructured texts cannot be searched and retrieved using deterministic criteria, as is commonly done when working with structured databases (Manning et al., 2008). Instead, information in unstructured texts is embedded in a stream of natural language, making it difficult to find relevant content in a targeted manner. The inherent variability and ambiguity of natural language further add to the complexity of processing text for IR systems (Krovetz & Croft, 1992).

One of the many difficulties that arise when working with unstructured texts is dealing with geographical information. In text documents, geographical references are often described using toponyms and spatial language (e.g. *'near Zurich'*) (Leidner & Lieberman, 2011). This presents a fundamental challenge for traditional IR systems that understand geographical references as concepts described by keywords or semantic representations rather than real geographies (Larson, 1996). They lack the ability to incorporate geographical requirements in information searches explicitly, as is usually done with structured geodata in traditional geographic information systems (Machado et al., 2011).

Geographic Information Retrieval (GIR) extends traditional IR to address this issue (Larson, 1996; Jones & Purves, 2008). One of the objectives of GIR is to interpret the geographical context in text documents and to transform it into a geographical footprint (Cai, 2002). Representing the geographical context of documents in this way allows geographical scopes to be considered when searching for information and is the basis for several systems for spatial search of unstructured web content, such as Web-a-Where (Amitay et al., 2004), SPIRIT (Purves et al., 2007) or Frankenplace (Adams et al., 2015).

Translating geographical contexts expressed as text into geometric footprints requires mapping geographical references described in natural language to real geographies (Louwerse & Zwaan, 2009). In its simplest form, this could mean, for example, assigning the coordinates (47.37, 8.55) to the toponym *'Zurich'* in a text describing the city of Zurich in Switzerland. Together with other referenced locations in the text, these coordinates can be used to estimate the geographical scope of the document and create a footprint, for example, in the form of a bounding box (Larson & Frontiera, 2004).

However, automatically inferring geographical scopes of text documents presents several challenges. Jones & Purves (2008) describe the three main difficulties that GIR systems face when interpreting geographical contexts in unstructured texts:

The first challenge lies in locating geographical references within the text, which often appear as toponyms. This process involves not only identifying words potentially representing toponyms but also confirming their actual use in a geographical sense (Jones & Purves, 2008). Because words that look like toponyms may also appear in text without conveying any geographical meaning, which is referred to as geo/non-geo ambiguity (Amitay et al., 2004). For example, in German, the name of the city of Zug in central Switzerland can also be used to refer to a train. Metonymy is another example where toponyms are used for entities that are not locations (Leveling & Hartrumpf, 2008). For example, depending on the context, *'Zurich'* may refer to a football club or even a global insurance company. Although an association with the namesake location may exist in these cases, these references do not refer to actual geographical locations but to organisations or companies. In this regard, (Gritta et al., 2020) present a comprehensive taxonomy for toponyms, distinguishing between literal toponyms, referring to actual physical locations, and associative toponyms that are only associated with them. Whether or not toponyms in texts are used to refer to actual geographical locations can often be inferred from syntactic and semantic cues in the context. In the two sentences, *'They are playing against Zurich'* and *'They are playing in Zurich'*, for example, interpreting the prepositions preceding the toponym can help in determining which sense of *'Zurich'* is intended in the given context.

The second difficulty is the mapping of geographical references to their corresponding referents (Jones & Purves, 2008). Commonly, this is done by looking up toponyms in gazetteers (Buscaldi, 2011). These are geographical dictionaries that map toponyms to geographical locations, which can then be used to extract the geometries of referenced locations (Goodchild & Hill, 2008). The fundamental difficulty with this is geo/geo ambiguity, i.e. the fact that different geographical entities may have the same name (Amitay et al., 2004). For example, *'Zurich'* may refer to the city situated on the northern end of Lake Zurich in Switzerland, but it may also refer to the canton in which the city is located. Furthermore, there are numerous other places in the world known as *'Zurich'*, like a small town in northern Kansas in the United States or a district in the Dutch province of Friesland. Thus, resolving a toponym first requires geographically disambiguating it (Overell, 2011). To do this, contextual cues can be used to narrow down semantic and geographical scopes of potential referents (Jones & Purves, 2008). This could be, for example, occurrences of context words like *'city'* and *'Switzerland'*, which may provide the necessary information to identify the location being referred to.

The last difficulty in creating geographical footprints of documents relates to vague geographical terminology. This arises, for example, from the use of colloquial place names that may not have precise or consistent spatial delineations (Jones & Purves, 2008). An example of this is the *'Niederdorf'*, the north-eastern part of Zurich's old town. Although official boundaries are defined for this zone, the common understanding of the term often extends beyond these. Vague spatial language can further complicate the interpretation of geographical context (Schockaert & De Cock, 2007; Derungs & Purves, 2016). These are, for example, phrases such as *'near Niederdorf'* or *'an hour's drive north of Zurich'*. Interpreting vague geographical terminology is particularly challenging, as it not only demands a good understanding of the language but also requires shared knowledge about places, which often cannot be derived from the context (Vasardani et al., 2013).

While processing linguistically complex spatial expressions is an important part of creating precise document footprints, this work does not address this issue. Instead, it focuses on the first two challenges described. These are addressed through the process of geoparsing, which is presented next.

2.2 Geoparsing

An important step in determining the geographical scopes of documents is the identification and interpretation of geographical references in the form of toponyms. This task is generally referred to as geoparsing and typically involves a two-step process (Gritta et al., 2018b): the first step is toponym recognition (also geotagging), which entails identifying toponyms in the text. The second step is toponym resolution (also geocoding), which involves linking the identified toponyms with their corresponding referents. In the following, these two tasks will be described along with a summary of techniques used to approach them.

2.2.1 Toponym Recognition

Toponym recognition is generally considered a specialised form of named entity recognition (NER) (Jones & Purves, 2008). NER involves identifying and classifying words or groups of words that represent named entities, such as names of people, organisations or places (Nadeau & Sekine, 2007). Toponym recognition differs from general NER in that only named entities of location-based categories are to be identified. For this reason, off-the-shelf NER systems are often used in geoparsing pipelines, with the output filtered in a subsequent step to include only location-based categories (Hu et al., 2023b).

Approaches to toponym recognition can be categorised into lookup-based, rule-based and machine learning-based methods (Leidner & Lieberman, 2011). Lookup-based approaches match words against lists of toponyms, while rule-based approaches use handcrafted rules to identify toponyms. However, both approaches are often limited in their ability to use complex contextual cues effectively. As a result, modern methods for toponym recognition are mostly based on machine learning techniques, which offer greater potential for looking at words in their specific context and thus classifying them more accurately.

Machine learning-based methods use statistical algorithms to train models to predict probabilities for words being (geographical) named entities (Leidner & Lieberman, 2011). These models receive input features derived from the target word and its context, which can either be manually defined or automatically extracted using various machine learning techniques (Hu et al., 2023b). Based on these features, the model generates an output, which is used to make a prediction about the target word. That could be, for example, the probability of the target word being a named entity or a probability distribution over different categories of named entities.

The effectiveness of machine learning-based toponym recognition methods strongly depends on the quality and quantity of the training data used for training models. As such, they are limited by the fact that creating large amounts of training data can be very costly and that trained models generally exhibit problems with generalisability to new text domains (Purves et al., 2018). Nevertheless, machine learning-based approaches, especially those employing deep learning techniques, have proven to be superior to other NER methods (Hu et al., 2023b). They benefit from sophisticated semantic understanding capabilities that come with the use of modern model architectures for natural language processing, allowing for more precise consideration of context when categorising words. Furthermore, recent work has also shown great interest in exploring the use of generative large language models (LLM) for NER.

These approaches formulate NER as a zero- or few-shot task, for which, for example, an LLM is prompted with instructions to identify named entities in a presented text (Xie et al., 2023).

2.2.2 Toponym Resolution

After toponyms have been identified in the text, the next step is associating them with their corresponding geographical locations. This could mean, for example, tagging toponyms with geographic coordinates or attaching unique identifiers that link to corresponding entries in a knowledge base. As such, toponym resolution can be understood as a specialised form of entity linking. The main challenge here is geo/geo ambiguity, i.e. different locations sharing the same name. For example, searching the GeoNames gazetteer for exact matches for the toponym *'Zurich'* would yield a list of 14 entries for locations that can be referred to using this name. Thus, a primary goal of toponym resolution is to disambiguate toponyms by determining from a list of potential candidates the most likely location that the toponym refers to (Buscaldi, 2011).

Traditionally, toponym resolution begins by searching gazetteers to generate lists of candidate locations for toponyms. These are then assessed according to various criteria to finally select the most likely location (Leidner, 2007). To do this, gazetteers and other types of ontologies can also be used as sources of information in the disambiguation of toponyms. They often contain additional information about places, which can be included in the assessment of candidates. Approaches that make use of such resources are what Buscaldi (2011) calls knowledge-based. In its simplest form, this can mean, for example, specifying default interpretations for toponyms based on certain attributes. Default locations can be, for example, ones with the largest population, the highest administrative level or ones that are considered most relevant based on some corpus statistic (Leidner, 2007). The use of default interpretations assumes that locations are more likely to be referenced if they are more relevant to the context in which they occur. However, these strategies have the obvious disadvantage that any references to less prominent referents of toponyms will automatically be disambiguated incorrectly. Some knowledge-based approaches, however, also make use of contextual elements. Gazetteers may contain administrative-hierarchical information about locations, which can, for example, be used to evaluate linguistic containment qualifiers (Leidner, 2007). Using this information could, for example, help in disambiguating mentions like *'Zurich, Kansas'* or *'Zurich (US)'* as Zurich in Kansas, based on the provided hierarchy information `Zurich < Rooks < Kansas < United States`.

Map-based approaches are another group of methods for toponym resolution (Buscaldi, 2011). These methods are based on the assumption that geographical references within a particular discourse are spatially autocorrelated (Purves et al., 2018). This means, for example, that the two toponyms *'Topeka'* and *'Wichita'* (both names of cities in the state of Kansas) occurring in a text would make it more likely that a mention of *'Zurich'* in the same text would refer to the town in Kansas rather than to the city in Switzerland. In this way, locations for a set of toponyms can be determined by minimising a geometric distance function between location candidates. This can mean, for example, selecting the referents for all toponyms in a text in such a way that pairwise distances between locations are minimised (Leidner, 2007).

Finally, there are machine learning-based, also known as data-driven, approaches (Buscaldi, 2011). Similar to techniques used for toponym recognition, these methods use manually defined or automatically extracted features from the toponym's context, which are used to make predictions using a trained machine learning model. These may consist of other toponyms occurring in the text or other kinds of textual elements, like context words potentially providing clues pointing to certain geographical regions (Speriosu & Baldridge, 2013). For example, the word *'Sechseläuten'*, the name of a traditional spring festival in Zurich, Switzerland, can serve as a powerful hint in disambiguating an occurrence of *'Zurich'*. Machine learning models can be trained to associate occurrences of certain words or even specific combinations of words with geographies, which can, in turn, be helpful in identifying the correct locations (DeLozier et al., 2015).

Machine learning-based toponym resolution systems typically produce rankings, classifications or regressions (Zhang & Bethard, 2024). Ranking-based systems take a list of location candidates and compute scores for each candidate that can then be used to rank lists. To do this, candidates are typically specified as additional input for models, which then generate candidate-specific scores. Classification systems, on the other hand, predict locations for toponyms by framing toponym resolution as a multi-class classification task, where the classes correspond to a discretised map of the earth's surface. In this way, models are trained to associate textual inputs with specific classes, i.e. regions on the earth, and thus learn geographical distributions of textual input, such as words, combinations of words or entire text segments (DeLozier et al., 2015). Regression approaches are very similar to those of classification, with the difference that instead of classifying, models perform multivariate regression, in which continuous values, typically geographical coordinates, are predicted.

Machine learning-based toponym resolution approaches also suffer from the fact that they are highly dependent on the availability and quality of training data. Training data is required for which toponyms have not only been identified but for which the locations they refer to have also been determined. Producing such training corpora demands extensive geographical knowledge from annotators and is a very laborious and costly task (Karimzadeh & MacEachren, 2019). The availability of such datasets, especially across diverse domains, is, therefore, one of the main challenges in developing machine learning-based toponym resolution methods (Gritta et al., 2018b; Hu et al., 2023a).

Despite this obstacle, developments in toponym resolution methodologies have increasingly relied on deep learning techniques (Zhang & Bethard, 2024). Modern deep learning architectures for natural language processing have created new opportunities for more sophisticated processing of text, which is a crucial requirement for improving the quality toponym resolution (Gritta et al., 2018b; Purves et al., 2018). Particularly influential in this regard was the introduction of the transformer architecture (Vaswani et al., 2017). It is the basis for models like BERT (Bidirectional Encoder Representations from Transformers), which allows for words and phrases to be analysed in the context of their entire sentence or document (Devlin et al., 2019). Their capability of capturing both syntactic and semantic nuances in texts offers the potential for improved handling of ambiguities in geographical references and may ultimately lead to more precise resolution of geographical references. For this reason, recent approaches to toponym resolution have increasingly implemented transformer models to better account for the context in which toponyms are used. The following section will briefly introduce transformers and how they can be used for toponym resolution.

2.3 Transformer

In recent years, transformers have led to remarkable advances in natural language processing. They have stood out due to their ability to capture contextual relationships across large bodies of text and use them to extract deeper and more nuanced meanings for specific tasks (Vaswani et al., 2017). There are different variations of the transformer architecture, which are tailored for different applications and requirements. In this work, the focus will be on encoder models, such as BERT, which are typically used when tasks involve understanding and interpreting text and making predictions based on it, as in word or text classification tasks (Devlin et al., 2019). This is in contrast to decoder models, which are typically used for text generation (Radford et al., 2018), and encoder-decoder models, which are used in sequence-to-sequence tasks, such as machine translation (Vaswani et al., 2017). In the following, the basics of encoder transformers and the contextualised embeddings they generate are presented.

Encoder transformer models typically accept text of variable length as input and generate individual vector representations for each element in the input sequence (Devlin et al., 2019). In this sense, inputs are often considered sequences of subwords or tokens, rather than normal words, as transformer models usually split texts into units smaller than words to limit the size of the vocabulary (Wu et al., 2016).

Vector representations of words or tokens are often referred to as embeddings. Embeddings model the semantics of words by projecting them into a vector space in which words of similar meanings are represented through similar vectors (Jurafsky & Martin, 2024). Semantic vector spaces are based on the distributional hypothesis, which states that words with similar meanings tend to be used in similar contexts (Harris, 1954). Creating these embeddings thus involves using statistics derived from text corpora that reflect the contexts in which words are used.

For natural language processing, embeddings provide the great advantage of modelling the abstract concept of meaning in a mathematical vector space. As words with similar meanings will also have similar vectors, semantic similarities between words can be calculated using simple distance metrics, such as the scalar product of their vectors or the cosine of the angle between them (Jurafsky & Martin, 2024).

Traditionally, word embeddings, such as those generated using Word2vec (Mikolov et al., 2013), have been limited by the fact that they are static. This means that a word would always be represented by the same vector, regardless of the context in which it is used. Static embeddings are the product of every possible context in which a word is used in a corpus without considering any potential ambiguities. They represent a conglomeration of all the different meanings a word can have and thus fail to reflect the specific meaning words convey in different contexts.

In contrast, embeddings generated by transformer models are dynamic, which means that they vary depending on the context in which words are used. This is made possible by the self-attention mechanism of the transformer architecture, which allows each word in a text to be considered in relation to all other words (Vaswani et al., 2017). During a comprehensive pretraining, transformers are taught to weigh and judge relationships between words using specialised pretraining tasks. They can then use this ability when creating word embeddings to varyingly consider context words according to the relevance they have for the target word (Devlin et al., 2019).

Thus, embeddings generated by transformer models are different depending on the context in which words are used and are, for that reason, often also referred to as contextualised embeddings (Jurafsky & Martin, 2024).

Typically, transformer embeddings are used as input for further machine learning models, which are trained to make predictions based on them. This could be, for example, the aforementioned named entity recognition, where the task may consist of predicting whether individual words are named entities. This could be implemented, for example, using a neural network that takes the contextualised embedding of a target word as input and is trained to generate an output that reflects the probability of the word being a named entity. The encoded information in the embedding about how the word was used allows the model to interpret the specific meaning of words more precisely and ultimately make more accurate predictions.

These capabilities of transformers offer promising possibilities for tasks that require a precise distinction between different meanings of words, as is the case with toponym resolution. Contextualised embeddings of toponyms allow the modelling of their relations to other words in the context, which could provide important clues for their disambiguation. The adaptability of transformer models through task-specific fine-tuning could further enhance these capabilities by drawing their attention to the specific linguistic features through which geographical information is expressed. The following chapter will present how transformers can be used for the task of toponym resolution based on exemplary implementations.

3 Related Work

Efforts to make use of transformer models for toponym resolution have led to a variety of approaches. At the time of writing, eight publications have been identified that implement transformer models in toponym resolution methodologies. At the same time, general entity linking systems have also increasingly implemented transformer-based techniques. Hu et al. (2023a) have shown that some of these systems were able to achieve state-of-the-art performance in the task of toponym resolution, even outperforming some specialised toponym resolution systems. For this reason, the following overview also includes the two best-performing entity linking systems evaluated by Hu et al. (2023a) for toponym resolution: BLINK (Wu et al., 2020) and GENRE (De Cao et al., 2021).

3.1 Transformer-based Toponym Resolution

The presented approaches can be categorised into three main groups: ranking-based approaches, localisation-based approaches and generative approaches (Table 1). In the following, the general workings and strategies of each of these groups will be presented:

Ranking-based approaches use transformer models to rank lists of location candidates for each toponym based on their likelihood to be the correct location. Typically, bi-encoder (Halterman, 2023; Li et al., 2023) or cross-encoder (Wu et al., 2020; Zhang & Bethard, 2023) strategies can be applied for this purpose. In bi-encoder approaches, separate vector representations are created for the toponym in its context and for each location candidate. Similarities between these representations can then be computed to rank candidates accordingly. This approach has the advantage of being computationally efficient, as toponyms only need to be encoded once, and representations of candidates can be precalculated and reused. Cross-encoder approaches, on the other hand, combine the toponym and each candidate in individual concatenated inputs to generate scores for each toponym-candidate pair. This allows for deeper interaction between the toponym and individual candidates but is much

Reference	Description
	Ranking-based approaches
Halterman (2023)	Mordecai 3 uses the transformer-based NER model from spaCy to recognise toponyms and then reuses the created embeddings for similarity scoring of location candidates from GeoNames.
Li et al. (2023)	GeoLM is a transformer model trained using contrastive learning to generate geospatially grounded representations of both toponyms and geographical entities to rank location candidates based on their cosine similarity to the toponym.
Wu et al. (2020)	BLINK first uses a transformer-based bi-encoder to create semantically relevant candidate lists based on similarity scores of vector representations and then employs a cross-encoder for a more precise ranking of the resulting list.
Zhang & Bethard (2023)	GeoNorm uses a transformer model to generate representations of toponym-candidate pairs, which are used to rank candidates in a two-stage process using custom neural networks.
	Localisation-based approaches
Cardoso et al. (2022)	A BERT model is used to generate embeddings of toponyms with different amounts of context, which are then passed through LSTM units to predict the most likely geographic region using classification.
Radford (2021)	ELECTRo-map uses a transformer model to encode text documents containing geographical references and perform multivariate regression to predict a probability distribution for geographic coordinates.
Solaz & Shalumov (2023)	An encoder-decoder transformer model is used to translate texts with toponyms into sequences of hierarchical cell encodings, which are then used to predict the most likely geographic cell.
	Generative approaches
De Cao et al. (2021)	GENRE uses an encoder-decoder transformer model to translate entity mentions in texts into unique textual representations in the form of Wikipedia article titles, which are used to link toponyms to corresponding entries on Wikipedia.
Hu & Kersten (2024)	A Large Language Model is used to generate unambiguous geographical descriptions for marked toponyms in a given text, which are then converted into coordinates using a geocoding service.
Zhang et al. (2024)	GeoPLACE uses BERT's masked language modelling capability to generate likely geographic attributes for toponyms in texts, which are used to filter candidate lists and identify the referenced location.

Table 1: Overview of transformer-based toponym resolution approaches

more computationally intensive because a separate model inference is required for each candidate, along with the entire document context of the toponym.

Localisation-based approaches aim to directly predict the geographical position of a toponym without relying on lists of candidates. These methods use contextualised representations of toponyms generated by transformer models to infer spatial information from text. This is done by formulating toponym resolution as either a classification (Cardoso et al., 2022; Solaz & Shalumov, 2023) or regression (Radford, 2021) problem. Classification approaches divide the earth's surface into discrete cells or regions. A model is then trained to predict the most likely cell in which the referenced location is located based on the embedding of the toponym and its context. Similarly, regression approaches aim to predict direct coordinates from embeddings. This is done by training models to estimate the geographical coordinates of the referenced location. In many cases, candidate lists are still used in localisation-based approaches to ultimately link predictions to knowledge bases by selecting the location that best matches the predicted geographical position.

Finally, generative approaches leverage the text-generative capabilities of certain transformer architectures for resolving toponyms. Instead of scoring candidates or directly predicting coordinates, models are trained to generate text sequences for toponyms within a provided document that can be used to uniquely identify the referenced location in a database. Generated texts can be, for example, a list of administrative hierarchies (Hu & Kersten, 2024), geographical attributes (Zhang et al., 2024), or even a descriptive identifier of the corresponding entry in a knowledge base (De Cao et al., 2021). These texts are finally used to match toponyms with their corresponding entries in a gazetteer by using the generated information to identify the specific locations.

3.2 Research Gaps

The analysed methods revealed several research gaps in the application of transformer models for toponym resolution. First, a critical limitation of some works concerns the lack of task-specific fine-tuning of transformer models. For example, the approach of Cardoso et al. (2022) uses a transformer model for encoding texts without first fine-tuning it for the specific task. Likewise, for Mordecai 3 (Halterman, 2023), the transformer model was not fine-tuned for the specific similarity comparisons used to rank candidates. The lack of specialised fine-tuning for toponym resolution tasks could mean that the models may not be able to effectively identify and use the relevant features in the text. Without fine-tuning, the transformer models simply create generic text representations instead of ones that explicitly capture the relevant geographical relationships within the text.

Another issue concerns the incorporation of linguistic context into the input for transformer models. Two works explicitly chose not to incorporate broader linguistic context and instead only consider geographical information in the form of other toponyms in the document (Zhang & Bethard, 2023; Zhang et al., 2024). The authors argue that the limited input size of transformer models would limit the usefulness of incorporating linguistic context while considering all co-occurring toponyms within the document would be more important for disambiguating toponyms. However, such a strategy disregards any contextual clues not presented as toponyms, which could severely limit the extent to which toponyms can be interpreted within their context.

Third, the computational resources required to run transformer models can be very substantial. Particularly, ranking-based cross-encoder approaches, such as those used in the systems BLINK (Wu et al., 2020) and GeoNorm (Zhang & Bethard, 2023), can be very computationally expensive and difficult to scale. In these approaches, separate transformer representations need to be generated for each combination of toponym and candidate, which can quickly accumulate large numbers of model inferences. A much more efficient approach is the use of bi-encoder architectures, as implemented, for example, in GeoLM (Li et al., 2023). For these systems, the representations of toponyms and candidates are generated independently of each other and only compared afterwards, which makes it possible to pre-compute and cache the representations of candidates. This means that only a single model inference would have to be carried out for each toponym to be resolved, which substantially reduces the required resources when disambiguating toponyms.

Finally, evaluating and comparing the performance of different toponym resolution systems is a challenging task. Although there appears to be a general agreement on which datasets and evaluation metrics to use for assessing systems, the comparability of results across different studies still presents a challenge. Differences in evaluation processes often lead to substantial discrepancies in the reported performance results, making it difficult to fairly compare methods with each other based on published results. There is, therefore, a strong need for a more clearly defined evaluation framework for toponym resolution systems to improve the transparency and reproducibility of evaluations and enable a more precise assessment of progress in the field.

4 Method

The aim of this work was to explore an efficient and easily adaptable use of transformers for the task of toponym resolution. To this end, a new method was developed, which is presented in this chapter.

The proposed methodology takes a ranking approach, using a bi-encoder architecture to compare toponyms in their context with potential location candidates and rank them based on their relevance. This approach was implemented using the SentenceTransformers framework introduced by Reimers & Gurevych (2019), which is normally used to determine semantic similarities between texts. This framework was adapted so that it can be used to efficiently determine the similarity between contextualised toponyms and their candidates.

Furthermore, the Geoparser[1] Python library developed as part of this work is presented, which integrates the proposed method into a complete end-to-end geoparsing pipeline. The library uses the NER functionality of spaCy[2] for toponym recognition and an adapted SentenceTransformer[3] model for toponym resolution. Geoparser is designed to be easily customisable by users to meet the specific requirements of different text types and domains.

The following sections of this chapter first describe the developed toponym resolution method and then present the architecture and functionalities of the Geoparser Library.

4.1 Proposed Method

At the heart of the proposed method lies the idea of treating the task of toponym disambiguation as a special form of text similarity estimation. For this purpose, the SentenceTransformer framework was adapted to facilitate an efficient use of trans-

[1] `https://github.com/dguzh/geoparser`

[2] `https://spacy.io/`

[3] `https://www.sbert.net/`

formers optimised for comparisons through the framework's bi-encoder architecture. In the following, the main elements of this approach are described.

4.1.1 Bi-encoder

Transformer models have established themselves as a powerful tool for processing natural language. However, the application of these models also comes with high computational demands, which significantly limits the scalability of transformer-based approaches. To address this problem, the proposed method employs a transformer model using a bi-encoder architecture.

Bi-encoders are particularly suitable for applications in which many comparison operations must be carried out, offering an efficient alternative to the conventional cross-encoder architecture. Cross-encoders process objects to be compared jointly through the model, which can result in a high number of model inferences. Bi-encoder systems, on the other hand, allow objects to be encoded separately into individual vector representations. The resulting vector representations can then be compared with each other in a computationally efficient way, for example, by calculating cosine similarity scores. This leads to a substantial reduction in the required computing power, as vector representations of objects that have been generated once can be reused for unlimited comparisons.

For the proposed method, a bi-encoder approach is used to compare toponyms with potential location candidates. This entails mapping both toponyms and candidates into a shared vector space. Doing that allows vector representations of toponyms and candidates to be compared with each other so that their similarities can be determined mathematically. In this way, the candidate with the highest similarity can then ultimately be determined as the most likely location to which the toponym refers in the given context. The proposed approach builds on similar work that also makes use of bi-encoder strategies. GeoLM (Li et al., 2023) employs a bi-encoder architecture in a similar way with the goal of spatially aligning representations produced by a language model through contrastive learning. However, it was not optimised for the task of toponym resolution and, therefore, achieved very poor results in this context.

In contrast, the method presented here adopts a bi-encoder-based approach specifically designed for the task of disambiguating toponyms. For its implementation, the SentenceTransformers framework is adapted, a tool usually intended for computing semantic similarities between texts. The intention behind using this framework is to inherit the comparison capabilities of SentenceTransformer models and, at the same time, benefit from the highly useful features of the SentenceTransformers library for implementing and fine-tuning models intuitively.

4.1.2 SentenceTransformers

The SentenceTransformers framework offers a highly efficient approach to determining semantic similarities between texts. It is typically used in applications such as sentence similarity scoring, document clustering or semantic searches within document collections. The framework is based on a special form of the bi-encoder architecture called a Siamese network architecture. Siamese networks are characterised by the fact that a single model is used to encode both elements to be compared.

Determining the similarity between texts involves first independently converting the texts into embeddings using the specially trained SentenceTransformer model. This model is set up in such a way that embeddings of semantically similar texts end up near each other in the embedding space, while those of dissimilar texts remain further apart. This property of SentenceTransformer embeddings means that semantic similarities between texts can efficiently be computed using simple vector-based distance metrics such as cosine similarity.

The ability of SentenceTransformer models to represent semantically similar texts in a similar way is the result of extensive pretraining. During this pretraining, models are trained using contrastive learning to create embeddings for pairs of texts in a way that models how their meanings compare to each other. For positive pairs of texts, i.e. texts that are considered semantically similar, the model is tuned to produce embeddings that are close to each other. For negative pairs, i.e. texts that do not share similar meanings, the model is tuned to produce embeddings that are distant from each other. Through this training process, the model acquires the ability to capture the meaning of texts and model it in the form of embeddings.

To use SentenceTransformer models for comparing toponyms with candidates, both the toponyms within their contexts and the candidates must first be represented as texts. For the toponym, this is simply done by extracting the relevant text passage surrounding the toponym. For the candidates, on the other hand, textual representations need to be artificially constructed using information extracted from a knowledge base. For this, attributes of location candidates, such as their name, country, and other relevant geographic or administrative features, can be used to craft a sentence that describes locations in a linguistically coherent form.

Transforming candidates into texts allows them to be compared with toponyms in the same way as would be done with two regular texts. However, the aim of the comparison is no longer to evaluate the general semantics of the texts but rather to determine how well the geographical information embedded in the context of the toponym matches the attributes of different candidates. To enable the SentenceTransformer model to effectively adapt to this new task, targeted fine-tuning is necessary.

4.1.3 Fine-Tuning

Fine-tuning the SentenceTransformer model is a crucial step in adapting it for the specific task of disambiguating toponyms. It is performed through contrastive learning, similar to the initial pretraining, but with specially constructed training data. For this purpose, a geographically annotated text corpus is used, which consists of texts in which toponyms are identified and linked to their correct geographical locations in a database. For each annotated toponym, positive and negative training examples are generated in the form of pairs of texts. Positive training examples consist of pairs of toponyms and their correct candidate locations, whereas negative examples consist of toponyms paired with incorrect candidates.

During fine-tuning, the weights of the model are adjusted so that it learns to produce similar embeddings for toponym-candidate pairs referring to the same location but dissimilar ones for pairs referring to different locations. This process should force the model to attend to geographically relevant cues in the context of the toponym and to align them with geographical indicators in the textual representations of locations candidates. The fact that toponyms are only contrasted with candidates sharing the same name should help the model discern meaningful contextual cues rather than simply matching place names.

After the SentenceTransformer model has been fine-tuned, it can be used to generate embeddings for both toponyms within their context and textual representations of location candidates. The similarity between toponym and candidate embeddings is then finally used to rank location candidates, with the most similar candidate selected as the most likely geographical referent for the toponym.

4.2 Geoparser Library

The Geoparser library was developed to provide a complete end-to-end geoparsing pipeline that integrates the proposed toponym resolution method. The library uses the NER functionality of spaCy for toponym recognition, a gazetteer integrated as an SQLite database for generating candidate lists, and a fine-tuned SentenceTransformer model for toponym disambiguation. The aim of this library is to provide a flexible and easily adaptable platform that allows users to customise the choice of models and knowledge bases for their individual requirements and to optimise models for specific text corpora. In the following, the main components and functions of Geoparser are presented. All descriptions refer to the latest version of the library at the time of writing (0.1.8). Future versions may differ in functionality and features from those described here.

4.2.1 Toponym Recognition Module

The processing pipeline of Geoparser starts with the input of texts in the form of strings, which are preprocessed by an integrated spaCy NLP pipeline. Users can choose between different spaCy models when instantiating Geoparser, which support different languages as well as varying model sizes. Larger models usually provide more accurate results but are more computationally intensive, while smaller models are faster but potentially less accurate.

Preprocessing with spaCy involves several steps, including tokenisation, named entity recognition, and finally, representing texts in specialised data containers. The Geoparser library extends the default spaCy pipeline by introducing specialised data structures developed specifically for geoparsing. This means that texts are converted into customised GeoDoc objects instead of regular spaCy Doc objects. GeoDoc objects behave the same way as native spaCy Doc objects but integrate additional features for handling toponyms. One of these features is the filtering of named entities recognised by spaCy to keep only location-related entities. This forms the step of toponym recognition, for which toponyms are entities that have been categorised by spaCy as LOC (Location), GPE (Geopolitical Entity) or FAC (Facility).

The individual toponyms identified in a GeoDoc are represented as GeoSpan objects. Similar to the GeoDoc class, GeoSpan is a customised data structure based on a regular spaCy Span that has been extended with specific functionalities. These extensions provide the framework for the interactions between text documents, transformer models and knowledge bases that are required for the subsequent step of toponym resolution.

4.2.2 Toponym Resolution Module

Once toponyms have been identified in texts, the next step involves linking them to their corresponding geographical locations. This process begins with generating lists of potential location candidates for each toponym. To do this, Geoparser uses an integrated gazetteer implemented as an SQLite database. The current version of the library uses the GeoNames gazetteer by default, which contains the names of locations worldwide along with location attributes such as administrative hierarchical parents, population size or geographical coordinates. However, the design of Geoparser is extensible, so additional gazetteers can be easily incorporated in the future to better meet the geographical and thematic requirements of different text domains.

Candidate lists are generated by querying the database using the recognised toponym. This prompts a full-text search across primary and alternate names of locations in the database to identify candidate locations that can be referenced by the toponym. The query is constructed to only suggest candidates belonging to the group of best matches based on the degree to which tokens match between the toponym and the name or alternate name of the candidate. For example, searching for the toponym *'Zurich'* would return 14 candidates that can be primarily or alternatively referred to as *'Zurich'*, and thus, all represent a 100% match with the queried toponym. The candidate *'Zurich Airport'*, for example, would not be suggested here because with only one of two matching tokens, it only represents a 50% match and thus does not belong to the group of best matches. If no 100% matches are possible, the next best group of matches is considered. For example, searching for *'Risoux'* would return the two candidates *'Le Mont Risoux'* and *'Forêt du Risoux'*, which each match the query at 33% and form the group of best matches for this query. However, if the search was for *'Mont Risoux'* instead, these two candidates would have different match grades (66% and 33% respectively), and only the better of the two would be presented as a candidate. This greedy matching strategy allows the candidate generator to produce short lists while still resorting to partial matching if needed. The result of querying the database is returned as a list of IDs identifying the potential geographic referents of the toponym in the underlying gazetteer.

The next step involves converting both the toponym within its context and the location candidates into a textual form so that they can be processed by the SentenceTransformer model. For the toponym, the relevant text passage surrounding the toponym in the document is extracted and, if necessary, shortened to meet the input length limits of the model. In doing so, a custom truncation algorithm ensures that the toponym remains centred within the text segment, where possible, while also preserving the linguistic integrity of texts by only removing entire sentences from the beginning and end of the text. The candidates, on the other hand, are transformed into artificial but linguistically coherent sentences using attributes from the database. The aim of these textual representations is to describe locations using information that can be used to differentiate between candidates. These representations are generated according to a template, which is specified by the user based on the availability of attributes in the employed gazetteer.

After the texts have been prepared for both toponyms and candidates, a fine-tuned SentenceTransformer model is used to convert them into embeddings. In creating candidate embeddings, the candidates of all toponyms in the corpus are pooled to ensure that each location candidate is encoded only once. Consequently, this process becomes proportionally cheaper the larger the document collection to be processed is, as toponyms may recur in other documents. For future versions of the library, an optional caching functionality is planned that will allow users to store candidate embeddings in the database once they have been generated, further increasing the efficiency of geoparsing for subsequent operations.

Finally, the embeddings of toponyms are compared with those of their corresponding candidates by calculating the cosine similarity between them. In doing so, the candidate with the highest similarity is considered the most likely location referred to by the toponym in the given context. The selected location is linked to the toponym by storing the corresponding location ID in the toponym's GeoSpan object. This ID can later be used to retrieve further information about the location directly from the database, which forms the basis for further geographical analyses of the texts.

4.2.3 Training Module

A key feature of the Geoparser library is the ability to adapt SentenceTransformer models specifically for toponym disambiguation. These adaptations are facilitated by the GeoparserTrainer module, which provides an environment for fine-tuning these models. Fine-tuning begins with selecting a SentenceTransformer base model, which is loaded from the HuggingFace library. The training module is then fed with specially created training data, to train the model to create geographically discernible representations of toponyms and location candidates.

For this purpose, GeoparserTrainer loads geographically annotated text corpora that must be provided in a dedicated format. This format includes the start and end positions of the toponyms in the text, as well as the IDs that link them to the corresponding locations in the gazetteer. In the first step, GeoparserTrainer converts the corpora into the customised spaCy data structures, which transform documents into GeoDoc objects and annotated toponyms into GeoSpan objects. For every annotated toponym, GeoparserTrainer then generates positive training examples, in which toponyms are matched with their correct location candidates, and negative examples, in which they are matched with the remaining locations in the candidate list. These training examples are finally used to fine-tune the model with a contrastive loss function using the training functionalities provided by the SentenceTransformer library.

Currently, Geoparser provides two pretrained models of different sizes that have been fine-tuned for toponym disambiguation using English newspaper article texts. They offer a ready-to-use solution for using Geoparser, allowing geoparsing without any preparation. However, the toponym resolution quality when using these models to texts in languages other than English and potentially also to text types other than newspaper articles may be restricted due to the limited diversity of the training corpora used. For this reason, it might be useful for certain applications to fine-tune custom models from scratch using training data that is appropriate for the task at hand. To do this, users can access a variety of base models available through HuggingFace, allowing them to adapt the choice of model to individual requirements.

Finally, users may also choose to simply refine models that have already been fine-tuned for toponym disambiguation, allowing them to be optimised for specific domains or text types that differ from the original training corpus. To do this, new, domain-specific corpora are geographically annotated and then fed into Geoparser-Trainer for further fine-tuning. Since, in this case, models have already been fine-tuned for toponym resolution beforehand, the size of the training corpus can be substantially smaller than that required for the initial fine-tuning. This option thus provides a practical opportunity to optimise the performance of models for specific geoparsing tasks without the need for large amounts of training data.

5 Experiments

This chapter describes the experiments that were conducted to evaluate the performance of the Geoparser library. The experiments are divided into two main parts. In the first experiment, the individual components of the entire geoparsing pipeline are tested, i.e. toponym recognition, candidate generation and toponym disambiguation. In the second experiment, the ability of the transformer model to adapt to new domains is tested by training an already fine-tuned model using small subsets of different corpora and evaluating the impact on performance. For these experiments, the evaluation framework of Hu et al. (2023a) is replicated to allow a fair and direct comparison with existing systems.

5.1 Evaluation Framework

The experiments were designed to be compatible with the evaluation framework of Hu et al. (2023a), which provides standardised test datasets and an evaluation environment that allows the performance of different toponym resolution systems to be measured under identical conditions.[1] The framework comprises twelve geographically annotated English text corpora spanning different domains (Table 2). Each test set consists of several documents, each provided as a separate text file. A gold annotation file in JSON format is also provided for every dataset, containing the labelled toponyms for each document. Annotations include the start and end positions of toponyms in the text as well as associated geographical coordinates. In six of the twelve datasets, GeoNames IDs of the referenced locations are also labelled.

To evaluate a toponym resolution system using this framework, prediction files must first be created for each test dataset. For this purpose, a system first processes the text documents, after which the identified toponyms and the geographical coordinates determined by the system are written into the respective prediction files. The format of these prediction files matches that of the gold files, allowing a direct comparison between them. The evaluation script by Hu et al. (2023a) can then be used

[1] `https://github.com/uhuohuy/toponym-disambiguation-voting`

Dataset	Document Count	Toponym Count	Corpus Domain	GeoNames IDs
CLDW (Rayson et al., 2017)	62	34,713	Historic	No
GeoCorpora (Wallgrün et al., 2018)	6,648	3,100	Tweet	Yes
GeoVirus (Gritta et al., 2018a)	229	2,170	News	No
GeoWebNews (Gritta et al., 2020)	200	2,601	News	Yes
LGL (Lieberman et al., 2010)	588	5,088	News	Yes
NCEN (Ardanuy et al., 2022)	455	4,595	Historic	No
NEEL (Rizzo & van Erp, 2016)	4,078	481	Tweet	No
SemEval-2019-12 (Weissenbacher et al., 2019)	90	3,258	Scientific	Yes
TR-News (Kamalloo & Rafiei, 2018)	118	1,319	News	Yes
TUD-Loc-2013 (Katz & Schill, 2013)	152	3,852	News	Yes
WikToR (Gritta et al., 2018b)	5,000	25,242	Wikipedia	No
WOTR (DeLozier et al., 2016)	1,644	11,795	Historic	No

Table 2: Overview of Datasets provided by Hu et al. (2023a)

to compare prediction files with corresponding gold files, which computes various metrics and reflects how well the system predicted the coordinates for the toponyms in the documents. The script optionally accepts filter files, which can be used to filter toponyms before evaluating predictions, allowing comparisons between systems on uniform subsets of toponyms.

To compare Geoparser with other systems, the system results prepared by Hu et al. (2023a) are used. Although the authors tested 21 systems in their work, only for seven of them the complete sets of prediction files were available in their public repository. Because the prediction files of those systems are required to compare them using uniform subsets of toponyms, comparisons in this work will be limited to these seven systems.

The compared systems include the three general entity linking systems DCA by Yang et al. (2019), BLINK by Wu et al. (2020) and Bootleg by Orr et al. (2020), all three of which are based on neural architectures, with the latter two also incorporating transformer models. Three other systems are specialised toponym resolution systems. These are the adaptive context feature-based system by Lieberman & Samet (2012), the rule-based CLAVIN by Berico Technologies (2012) and the localisation-based deep learning system CamCoder by Gritta et al. (2018a). Finally, the voting system proposed by Hu et al. (2023a) is also used for comparison. This system uses spatial clustering of predictions from an ensemble of different systems to determine coordinates for toponyms.

5.2 First Experiment

Toponym Recognition

First, the toponym recognition component of the geoparsing pipeline is tested using the transformer-based spaCy model for English texts `en_core_web_trf`. The evaluation is conducted on all twelve test datasets of the evaluation framework, with the performance measured using the following metrics:

- **Precision:** the proportion of correctly recognised toponyms to all toponyms recognised by the system.
- **Recall:** the proportion of correctly recognised toponyms to all toponyms annotated in the gold files.
- **F1 Score:** the harmonic mean of precision and recall.

Toponyms are only classified as correctly recognised when the string boundaries of the recognised toponym exactly match those of gold annotations.

Candidate Generation

The second step in the pipeline involves generating a list of location candidates for each identified toponym. In this experiment, this was done once using all the annotated toponyms in the gold files and once using just the subset of these that were also identified as toponyms by spaCy. The lists of candidates were generated using a GeoNames database. Since the candidate generator produces lists of IDs that refer to the underlying database, its evaluation requires gold toponyms to be labelled with respective IDs. For this reason, only the six test datasets containing GeoNames IDs could be used for this evaluation.

Three metrics were used to measure the effectiveness of the candidate generator:

- **Coverage:** the proportion of toponyms for which the candidate lists are not empty.
- **Recall:** the proportion of toponyms for which the correct locations are included in the candidate list.
- **Mean List Length:** the average length of candidate lists.

Toponym Resolution

The next step consists of disambiguating toponyms by selecting the most likely location from the lists of candidates. This involved first preparing a SentenceTransformer model to be used for creating toponym and candidate embeddings. For this purpose, the English-language SentenceTransformer model `all-distilroberta-v1` was chosen as the base model to be fine-tuned. Training examples were created using the LGL corpus, as this is also the corpus that Hu et al. (2023a) used to retrain one of the compared systems (Adaptive Learning). Creating textual representations of location candidates is required for both the fine-tuning of the model and the disambiguation of toponyms. For this experiment, the following template was created to represent candidates using location information available in GeoNames: `[name] ([feature type]) in [admin2], [admin1], [country]`

To evaluate the toponym resolution component, two different prediction files were created for each of the remaining eleven test datasets. For the first file, all toponyms and their positions in the texts were directly extracted from the gold annotation files. This allowed an evaluation of the toponym resolution component without filtering the toponyms through the toponym recognition step. For the second file, however, the evaluation was carried out using only the sets of toponyms recognised by spaCy to give a more realistic picture of the performance of Geoparser when used as a whole pipeline. Because the provided prediction files of the comparison systems all contained the full set of toponyms, the respective sets of toponyms could be extracted to compare the systems on identical sets of toponyms.

Given the prediction files, the evaluation script by Hu et al. (2023a) computes the following performance metrics:

- **Accuracy@161km:** proportion of toponyms resolved within a radius of 161 km (100 miles) from the correct position.
- **Mean Error Distance:** average geographical deviation between predicted and actual positions.
- **Area Under the Curve:** error distribution of predictions, which is calculated by integrating the area under a curve of scaled logarithmic error distances (lower values indicate more precise predictions).

5.3 Second Experiment

The second experiment aims to determine the extent to which the performance of an already fine-tuned SentenceTransformer model can be improved for different domains by refining it using small amounts of additional domain-specific training data. For that, the model that was fine-tuned in the first experiment was further trained with small subsets of the remaining five test datasets that contain GeoNames IDs. Fine-tuning was only possible using these five datasets, as annotated GeoNames IDs are a requirement for creating training examples.

The training subsets were created in sizes of 100, 200, 300, 400 and 500 toponyms. To simulate a user annotating documents for creating training data, the training subsets were extracted at the document level. This involved randomly selecting whole documents until the target numbers of toponyms were reached. In doing so, larger subsets would always contain the toponyms of smaller subsets. This was done to gradually examine the effect of increasing amounts of training data on the performance of the model.

After every fine-tuning, new prediction files were created for all eleven test datasets to evaluate the performances of the new models. For the five datasets that were also used for fine-tuning, all the documents that were used to create the training subsets were removed from the test sets. This ensured that all models were tested on the same set of toponyms without overlap with the data used for training. For the remaining six datasets, the complete test sets were used. The evaluation was conducted using the same evaluation script as in the first experiment, measuring the performance using Accuracy@161km, Mean Error Distance and Area Under the Curve. For this second experiment, the evaluation was carried out only using the entire set of toponyms.

6 Results

6.1 First Experiment

Toponym Recognition

To evaluate the toponym recognition component of Geoparser, the ability of the spaCy model `en_core_web_trf` to recognise toponyms was tested using all twelve test datasets from the evaluation framework of Hu et al. (2023a). Performance was measured using precision, recall and the F1 score (Figure 1).

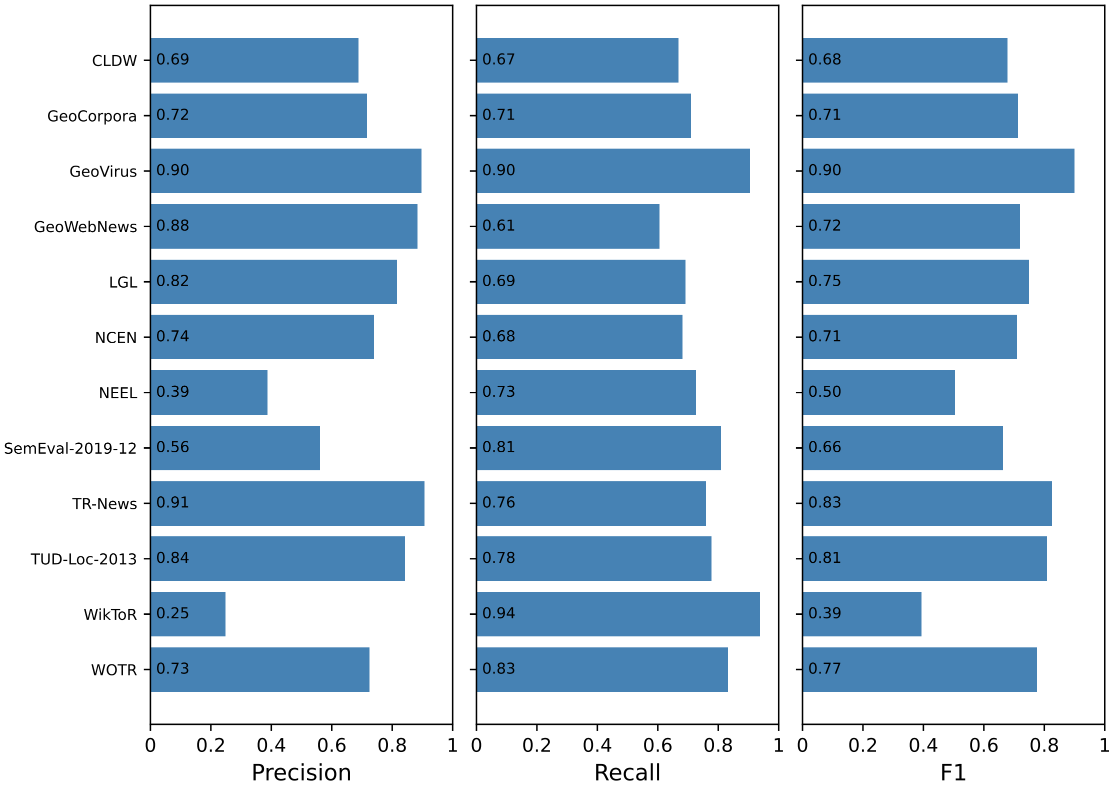

Figure 1: Toponym recognition performance

Precision varied considerably between datasets. The highest value was achieved for TR-News at 0.91, followed by GeoVirus at 0.90 and GeoWebNews at 0.88. For these datasets, most of the toponyms recognised by spaCy were also labelled in the gold annotations. The lowest values were measured for WikToR with 0.25 and NEEL with 0.39. For these, spaCy identified many toponyms that were not annotated as such in the datasets.

The highest recall of 0.94 was achieved for WikToR, followed by 0.90 for GeoVirus. For these datasets, spaCy was able to recognise most annotated toponyms. The lowest recall, in contrast, was measured for the datasets GeoWebNews (0.61) and CLDW (0.67).

The best overall performance, measured by the F1 score, was achieved for GeoVirus with 0.90, TR-News with 0.83 and TUD-Loc-2013 with 0.81. For WikToR and NEEL, the worst overall performances were obtained with F1 scores of 0.39 and 0.50, respectively, primarily due to the low recall values.

Candidate Generation

The evaluation of the candidate generator was carried out based on a GeoNames database using the six datasets that contain GeoNames IDs for annotated toponyms. Coverage, recall and mean list length were calculated for both the total set of annotated toponyms and the subset filtered by the spaCy toponym recognition step (Figure 2).

High coverage was achieved across all datasets, with slightly better coverage on the spaCy subsets compared to the full sets of toponyms. In five out of six datasets, the coverage was at least 0.97, with the only exception being the GeoCorpora dataset, where coverage was slightly lower, with 0.93 for all toponyms and 0.96 for the spaCy subset.

Recall was also high for most datasets, with consistent improvements when processing only the spaCy-recognised toponyms. Particularly high recall was achieved for TUD-Loc-2013, with 0.98 (all) and 0.99 (spaCy). For the GeoWebNews and GeoCorpora datasets, the recall was the lowest on the total sets of toponyms, with 0.77 and 0.84, respectively. A particularly large increase in recall was observed on the GeoWebNews dataset when considering the spaCy subset over the total set of toponyms, with an increase of 0.17 from 0.77 to 0.94.

The mean list lengths ranged from 31 to 46 candidates for all toponyms and from 32 to 51 for the spaCy toponyms. For all datasets, lists were, on average, slightly longer when only the spaCy subsets were processed. The longest average list length was measured for the LGL dataset with a mean list length of 46 (all) and 51 (spaCy),

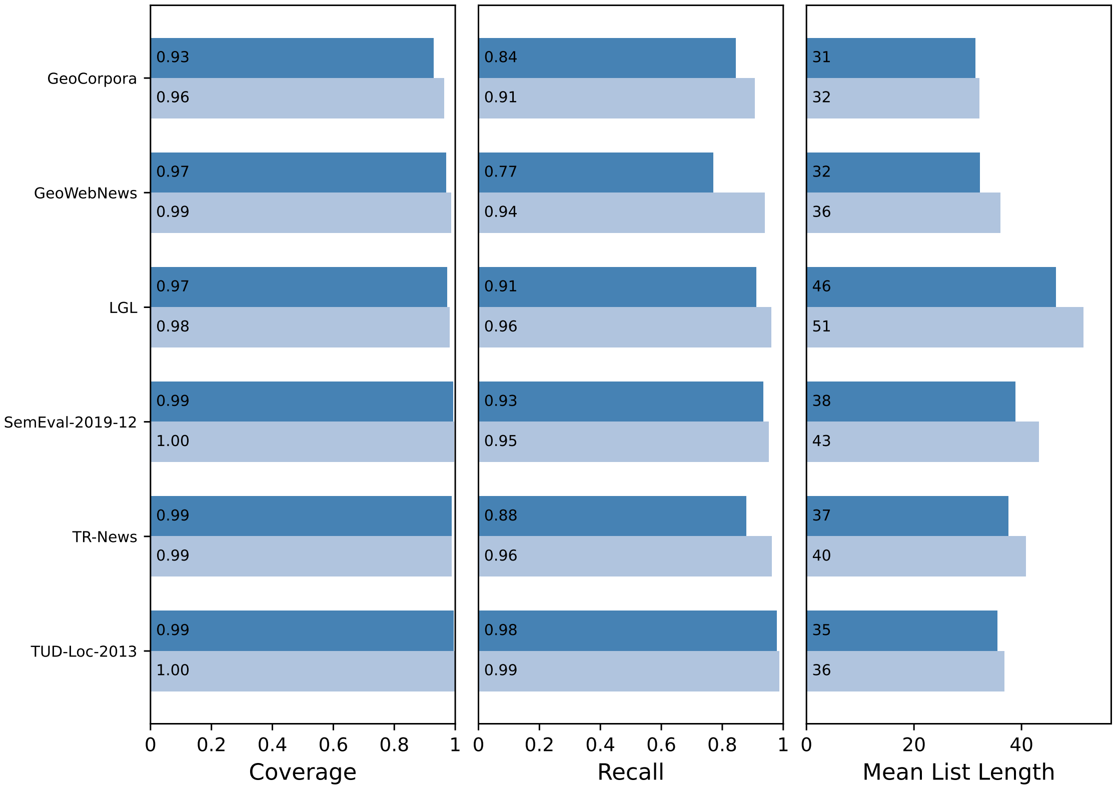

Figure 2: Candidate generation performance on all toponyms in the datasets (dark blue) and on the spaCy subsets (light blue)

while the shortest candidate lists with average lengths between 31 and 36 were generated for GeoCorpora and GeoWebNews.

Toponym Resolution

The evaluation of the toponym resolution stage was carried out based on the SentenceTransformer model `all-distilroberta-v1` that was fine-tuned using the LGL corpus. The model was tested on the remaining eleven datasets that were not used for fine-tuning. The tests were performed on the total sets of toponyms as well as on the subsets of spaCy-recognised toponyms. Performance in terms of Accuracy@161km (A161) is reported in Figure 3. The results for Mean Error Distance (MED) and Area Under the Curve (AUC) are provided in the appendix in Figure 5 and Figure 6, respectively.

The performance of Geoparser varied considerably between datasets. The best performance in terms of A161 was achieved for the datasets GeoVirus, TUD-Loc-2013, WikToR, TR-News and SemEval-2019-12, with values between 0.78 and 0.87. The worst A161 was achieved for CLDW and WOTR, with values ranging between 0.47

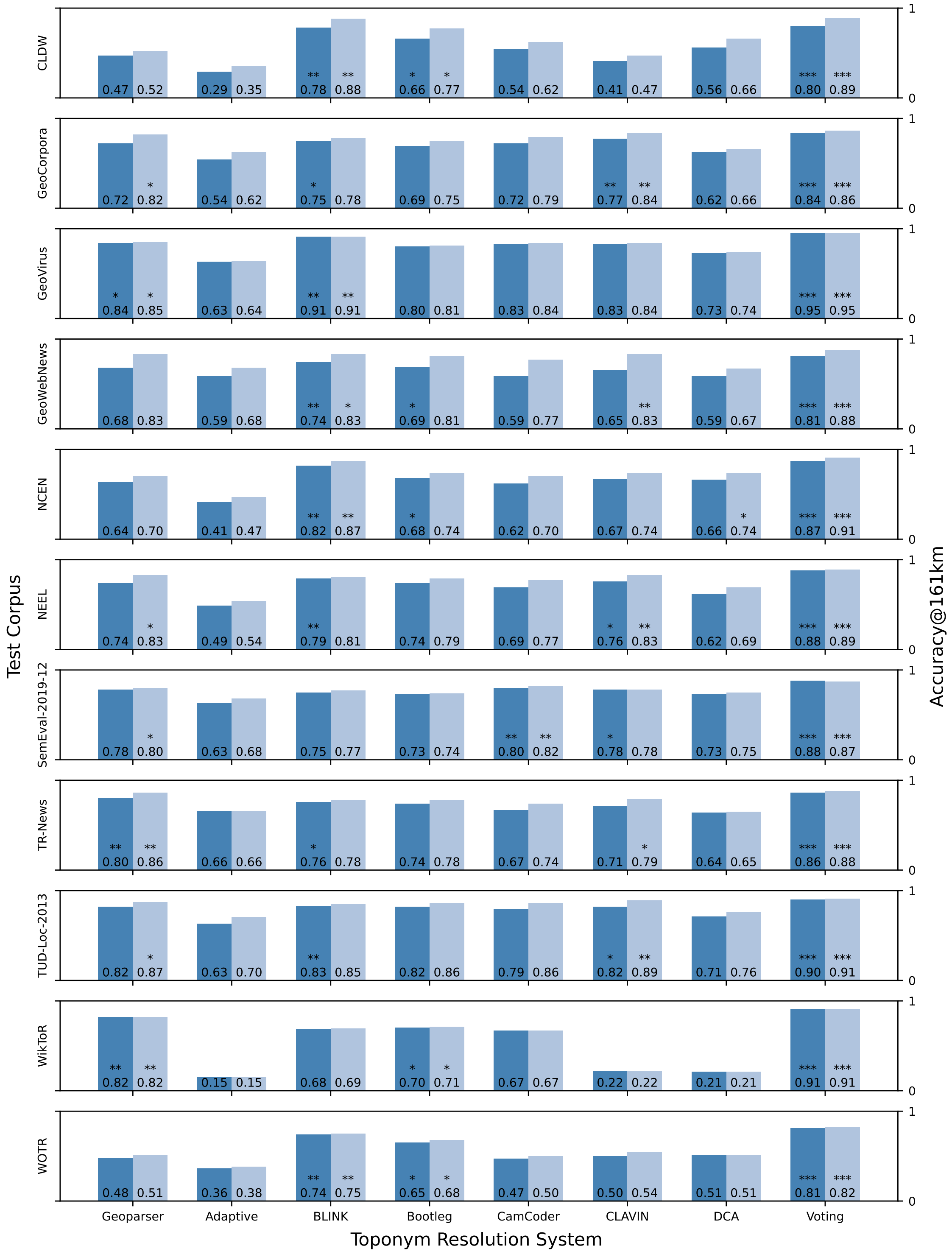

Figure 3: Toponym resolution Accuracy@161km on all toponyms in the datasets (dark blue) and on the spaCy subsets (light blue);
*** indicates the best system, ** the second-best, and * the third-best

and 0.51. Across all datasets, the performance on the spaCy subset was always equally good or better than on all toponyms. For the two datasets GeoWebNews and GeoCorpora, the differences between the total set and the spaCy subset were the greatest, with improvements of 0.15 and 0.10, respectively, when only toponyms recognised by spaCy were considered. When measured in MED, the results show a similar picture, with particularly low (good) values for the datasets GeoVirus and WikToR, with values between 460 and 637 km. The worst values were measured for CLDW, WOTR, and NCEN, with MED of over 4000 km on the total set of toponyms and over 3000 km on the spaCy subsets. Looking at the AUC again indicates similar patterns to those observed for the other metrics, with the lowest (best) values for the datasets TR-News, WikToR and TUD-Loc-2013 and the worst values for CLDW and WOTR. Particularly noticeable here are again the large improvements for GeoWebNews and GeoCorpora when only toponyms recognised by spaCy were processed.

Compared to the other seven systems, Geoparser achieved competitive performance on some datasets. Considering the A161, when all toponyms were processed, Geoparser ranked among the top three systems on three out of eleven datasets, reaching second place twice. Considering only the spaCy subsets, it achieved a top three ranking in seven out of eleven datasets, including second place twice. Looking at MED, Geoparser performed even better in comparison, achieving a top three ranking in ten out of eleven datasets when considering all toponyms and in nine out of eleven datasets for the spaCy subsets. In both cases, it came in second three times. In terms of AUC, Geoparser also achieved solid results: it was among the top three systems six times for both sets of toponyms, including five times among the top two for all toponyms and six times among the top two for the spaCy subsets. For two datasets, Geoparser was the best system in terms of AUC when all toponyms were considered, and for three datasets, it was the best system for the spaCy subsets.

Finally, the duration for fine-tuning the SentenceTransformer model using the LGL dataset and the runtimes for the toponym resolution process on all twelve datasets were measured. Both processes were carried out on an Ubuntu instance with a NVIDIA Tesla T4 GPU. Fine-tuning using the 5,088 toponyms in the training dataset took 2 hours and 49 minutes. For the toponym resolution, a total of 98,264 toponyms from 19,264 documents were processed. The total runtime was 55 minutes, of which 6 minutes were spent creating the candidate embeddings, 20 minutes creating the toponym embeddings and 29 minutes on the remaining processes such as context truncation, candidate generation and similarity calculations. Across all datasets, this resulted in an average of 29 resolved toponyms per second.

6.2 Second Experiment

To test the Geoparser's ability to adapt to different domains, the SentenceTransformer model that was fine-tuned for the previous experiment was incrementally trained with additional training examples from new text corpora. For this purpose, five additional datasets were used to create five training subsets of 100 to 500 toponyms each that were used for fine-tuning, resulting in 25 new models. The new models were then tested on the same eleven test datasets that were used for testing in the first experiment. The performance deviation for A161 is shown in Figure 4. Deviations in MED and AUC are provided in the appendix in Figure 7 and Figure 8, respectively. For the most part, deviations in A161, MED and AUC correlated, which is why the following sections do not differentiate between the three metrics.

The results of this experiment are distinguished into in-domain and out-of-domain cases. In-domain cases refer to the instances where the data used for fine-tuning originated from the same corpus as the test data, whereas out-of-domain cases refer to all other instances. The results of the five in-domain evaluations varied considerably. While in-domain fine-tuning for GeoCorpora and SemEval-2019-12 led to overall improvements, little to no changes in performance were observed for models fine-tuned with training examples from GeoWebNews and TUD-Loc-2013. For TR-News, there were minimal improvements after the first two subsets; however, performance was worse than the baseline model for the last three subsets.

Looking at all the tested datasets, there were large differences in the overall impact of additional fine-tuning. For the datasets GeoVirus, GeoWebNews and TUD-Loc-2013, there was little to no change in performance, regardless of which dataset or how many toponyms were used for fine-tuning. In other datasets, however, fine-tuning had a substantial impact on the performance of the model. For the WikToR and TR-News datasets, fine-tuning led to consistently worse performance across all datasets used for training. Particularly pronounced was the magnitude of the negative impact on performance on the WikToR dataset. For other datasets, such as CLDW, NCEN or NEEL, fine-tuning led to both improved and worsened performance, depending on the training corpus used and, in some cases, even on the specific training subset.

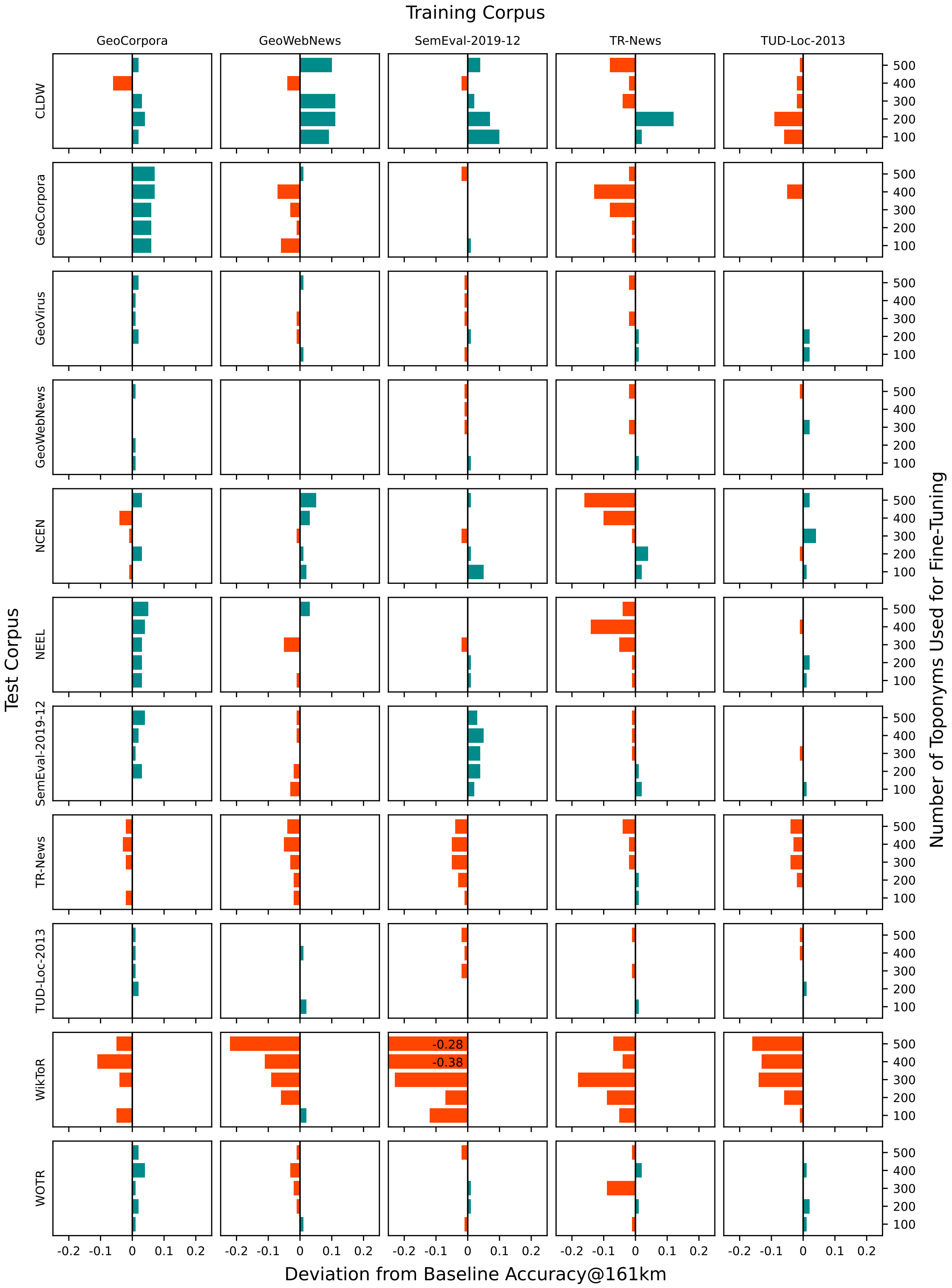

Figure 4: Accuracy@161km deviation from baseline model after further fine-tuning (model deterioration in orange and improvement in green)

7 Discussion

7.1 Interpretation of Results

7.1.1 First Experiment

Toponym Recognition

The first experiment involved evaluating the individual pipeline components of the Geoparser library. First, the toponym recognition stage was examined. According to a survey by Hu et al. (2023b) that evaluated the 27 most used toponym recognition systems on 26 test datasets, the best systems achieved an average precision of 0.85 to 0.87. For some of the datasets, the precision achieved by spaCy in the conducted experiments was similar to these values, and thus, solid results were achieved. However, for other datasets, such as WikToR, NEEL or SemEval-2019-12, precision was rather low. This is likely caused by different annotation strategies and schemes in the production of the datasets. For example, the very low precision for WikToR can be explained by the programmatic creation of the corpus. The authors collected texts based on geographically linked Wikipedia articles, which meant that each document would only be annotated with a single location (Gritta et al., 2018b). Another example is the SemEval-2019-12 dataset, which consists of scientific articles. The creators of the dataset chose not to annotate toponyms in the addresses of the authors of the articles (Weissenbacher et al., 2019). In an evaluation of a toponym recognition system processing the whole document, omitted annotations result in inherently low precisions.

For toponym recognition recall, the best systems tested by Hu et al. (2023b) achieved values between 0.70 and 0.78. These are comparable to the recall achieved by the evaluated spaCy model on most of the datasets. Again, some of the lower recall values could partially be attributed to different strategies used for annotating toponyms. For example, in the GeoWebNews, TR-News and LGL corpora, a large number of demonyms are annotated as toponyms, which spaCy categorises as NORP (Nationalities or Religious or Political Groups) and are therefore not considered as toponyms in the implemented version of Geoparser. Other reasons contributing to low recall can, however, be attributed to limitations of spaCy itself. For example, it was observed that spaCy sometimes was unable to extract toponyms from hashtags in documents from the tweet datasets GeoCorpora and NEEL, especially when they were not capitalised. Furthermore, spaCy sometimes incorrectly included the *'the'* article preceding toponyms like *'Middle East'* or *'Mediterranean Sea'*, which often resulted in a failure to match the annotation for the evaluation.

Given differences in annotation procedures and strategies for categorising terms as toponyms, it is difficult to compare performance across different datasets. Overall, however, the capability of the spaCy NER component as a toponym recognition module has been demonstrated to be robust and comparable to state-of-the-art systems.

Candidate Generation

Next, candidate generation was tested. Coverage was high for all six datasets, indicating that the candidate generator was able to successfully suggest potential locations for most toponyms. The only dataset where coverage was slightly lower was the tweet dataset GeoCorpora. This is likely due to the informal nature of the texts, for which the candidate generator is faced with matching unusually constructed toponyms with standardised names from the gazetteer. For example, difficulties were observed when matching irregular word forms extracted from hashtags, such as *'MississippiRiver'* or *'South #losangeles'*. Unusual or colloquial abbreviations such as *'Richland-ND'* for *'Richland County, North Dakota'* or *'SE DC'* for *'Southeast, Washington, D.C.'* also returned empty candidate lists for GeoCorpora. Overall, coverage was always slightly higher on the spaCy subsets than on all toponyms, likely because spaCy filtered out some of these special forms of toponyms.

The recall was measured to determine how often the correct referent was included in the generated candidate lists. Candidate generators are often evaluated using recall@n, assessing whether the correct candidate appears in the first n results from a ranked list of candidates. For example, Zhang & Bethard (2023) report values for recall@20 on the GeoWebNews, TR-News and LGL datasets for their Lucene-based candidate generator of the GeoNorm system, which were 0.87, 0.97 and 0.96, respectively. Since the candidate generator of Geoparser does not produce ranked lists, the recall was calculated based on the entire candidate list instead. On the same datasets, the recall of Geoparser on the entire sets of toponyms was consistently worse than that of GeoNorm, at 0.77, 0.88 and 0.91, but better than that of the other two systems evaluated by Zhang and Bethard, DeezyMatch (Hosseini et al., 2020) at 0.67, 0.70 and 0.54, and SAPBERT (Liu et al., 2021) at 0.75, 0.78 and 0.74.

A main source of problems for the candidate generator of Geoparser were demonyms, which occurred frequently in datasets like GeoWebNews, TR-News or LGL. For example, searching for *'African'* would return candidates such as *'African Banks'*, *'African Lake'* or *'African Jordan'*, but not the candidate *'Africa'* implied by the annotators, which constitutes a 0% match for the token-based full-text search engine underlying the candidate generator. Other instances in which the candidate generator often showed difficulty in recommending the correct location were abbreviations such as *'B.C.'* for *'British Columbia'*. If abbreviations were not recorded in the gazetteer, the candidate generator was unable to retrieve the implied referents and instead suggested candidates for which the abbreviation did occur in the name, such as *'Marungu B.C.'*. In other cases, however, the reason that the correct candidate was not found was that the annotations were simply inaccurate. For example, toponyms such as *'University of California'* would sometimes be annotated with the referent for *'California'* or *'Central Europe'* with the referent for *'Europe'*. In such cases, the candidate generator correctly retrieved candidates for the complete name, but the incorrect annotation meant that the result would automatically be categorised as faulty. Finally, in some instances, the annotated GeoNames IDs were no longer up to date, meaning that attempts to match candidates with the supposedly correct referent were categorised as errors, even if they would have been correct.

Finally, the length of the generated candidate lists did not vary greatly between the datasets, except for the LGL corpus, for which lists were, on average, a bit longer. The fact that toponyms in the LGL corpus resulted in longer candidate lists indicates that toponyms are potentially more ambiguous than those in other datasets. This is not surprising, given the way the LGL corpus was constructed. With the aim of making the dataset more challenging for geoparsing evaluations, the authors specifically aimed to include articles from newspapers based in places with highly ambiguous names, the idea being that the articles they publish would more likely contain ambiguous toponyms (Lieberman et al., 2010). Overall, candidate lists were observed to be longer for the toponyms in the spaCy subsets than when all toponyms were processed, suggesting that spaCy filters out toponyms that are either more difficult to query or just generally less ambiguous.

Toponym Resolution

Lastly, the complete toponym resolution component was tested. An important consideration in this evaluation is that the outcome of the embedding-based disambiguation is strongly influenced by the quality of the preceding candidate generation. Whenever relevant candidates are not presented for selection, the disambiguator will automatically fail to make correct predictions as it is forced to choose a candidate from a list where all of them are incorrect. The recall of the candidate generator thus forms an accuracy ceiling for the overall performance of the toponym resolution stage. This is illustrated, for example, by the results for GeoWebNews. For this dataset, a strong increase in performance from an A161 of 0.68 to 0.83 was observed when only the spaCy subset was processed compared to the entire set of toponyms. Much of this difference is likely attributable to the candidate generator, where recall was particularly low for all toponyms at 0.77 but relatively high at 0.94 for the spaCy subset. This suggests that spaCy filters out a large proportion of the toponyms contributing to low candidate generation recall and, in doing so, substantially improves the measured performance for the overall toponym resolution.

However, it is likely that there are also other factors contributing to the overall better performances on spaCy subsets, as is evident from the results for TUD-Loc-2013. For this dataset, there were also improvements in the toponym resolution performance on the spaCy subset compared to all toponyms, however, candidate generation recall was almost identical for both sets. This suggests that spaCy is likely also filtering out toponyms that are generally more difficult to resolve, independently of the ability of the candidate generator to suggest relevant candidates.

Overall, Geoparser was able to achieve competitive toponym resolution performance for most of the datasets. Exceptions were the three datasets CLDW, NCEN and WOTR, for which performances were very low. Given that all three of these corpora are composed of historical documents, a possible explanation for the poor performance on these datasets could be an inability of the SentenceTransformer model to generalise to these new domains. After all, to fine-tune the model underlying the Geoparser, the LGL corpus, which consists solely of newspaper articles, was used. Linguistic and structural differences between newspaper articles and historical texts could have affected the effectiveness of the model when processing documents from these new domains. Having said that, it is surprising that the performance on the tweet datasets GeoCorpora and NEEL did not suffer to a similar extent, given the large differences in both language and structure of tweets compared to news articles. While it is still likely that out-of-domain usage of the model contributed to the poor performance on the historical datasets, other factors are likely to have played a bigger part in it.

The bigger cause of the poor performance on the historical datasets likely relates to the gazetteer employed for candidate generation. GeoNames includes some historical toponyms and even defines dedicated feature types for them (e.g. historical political entity). However, the coverage of historical toponyms in GeoNames has been shown to be largely insufficient to meet requirements for spatial analyses of historical documents (Grover & Tobin, 2014). It is, therefore, likely that the GeoNames-based candidate generator of Geoparser could have had difficulties suggesting relevant locations for some of the historical toponyms. However, the fact that these datasets do not use GeoNames as a source for grounding toponyms also means the degree to which the gazetteer contributed to the poor performance could not be quantified. However, this claim is supported by the fact that systems like Adaptive, CamCoder and CLAVIN, which all also use GeoNames as a gazetteer, had similar performance ceilings on the historical datasets. The Wikipedia/Wikidata-based systems BLINK and Bootleg, on the other hand, appear to have performed better on these datasets, which could indicate that these knowledge bases may provide more comprehensive coverage of historical toponyms than GeoNames.

Another important consideration when interpreting the measured toponym resolution performances is the metric used to evaluate them. For example, Geoparser scored much better compared to other systems when evaluated in terms of AUC, compared to A161 or MED. The A161 is a metric that treats all predictions with an error distance of less than 161 km equally, regardless of how precise the predictions are within this threshold. Similarly, all error distances greater than 161 km are also considered equally. The AUC, on the other hand, differentiates between continuous error distances and also accounts for different magnitudes of error distances by logarithmising them. A particularly high AUC can mean that the system produced more precise predictions, even if overall, fewer predictions fell below the 161 km threshold. That is because the AUC strongly rewards more accurate predictions compared to even minor deviations, but can be very forgiving of major outliers. Extreme outliers can arise, for example, when the evaluation script assigns maximum error distances of 20039 km to missing predictions, as is the case for Geoparser when empty lists are produced during candidate generation.

Finally, the runtimes for the toponym resolution process on the complete sets of all twelve datasets were measured. This was done to get a general impression of the efficiency of Geoparser and to compare it with existing transformer-based systems. Unfortunately, a proper comparison could not be made as the systems to be compared were not operated on the same platform used for running Geoparser.

However, an approximate comparison could be made with the runtimes measured by Hu et al. (2023a) using the same twelve datasets. For the transformer-based systems BLINK, Bootleg, GENRE, ExtEnD and LUKE, they used a NVIDIA Tesla V100 GPU and measured runtimes of 40.4h, 3.2h, 22.6h, 3.5h and 6.5h. In comparison, Geoparser only required 0.9h to complete the same task using a NVIDIA Tesla T4. Considering that the GPU used for running Geoparser has only half the number of CUDA cores of the GPU used by Hu et al. (2023a), it is possible that Geoparser would have performed even better in a direct comparison.

7.1.2 Second Experiment

The second experiment aimed to investigate how well the Geoparser could be optimised for specific domains by further fine-tuning it on small amounts of data from different corpora. The baseline model for this experiment was the model from the first experiment, which was trained on the complete LGL corpus of newspaper articles. It was then further fine-tuned using subsets of toponyms from five additional datasets, consisting of tweets (GeoCorpora), scientific articles (SemEval-2019-12), and news articles (GeoWebNews, TR-News and TUD-Loc-2013).

It was observed that further fine-tuning with data from new domains led to improved performances on datasets from these respective domains. For example, fine-tuning with subsets of the tweet dataset GeoCorpora led to improvements on the test set of GeoCorpora itself and on the other tweet dataset NEEL, with increases in A161 of up to 0.07 and 0.05, respectively. Similarly, further fine-tuning using training data from SemEval-2019-12 led to improvements in A161 of up to 0.05 on the test set of that same corpus. Additional training with toponyms from the remaining three datasets, which all consist of newspaper articles, on the other hand, resulted in little or no change in performance across most of the other newspaper article corpora. This suggests that further training on domains differing from the original training data indeed improves the quality of toponym resolution for these domains. In doing so, the size of the subset used for additional training seems to be of limited importance, given that improvements in performance did not grow proportionally with the amount of training data used.

In some cases, additional training also led to drastic changes in performance on other datasets. For example, it is unclear why the quality of toponym resolution on the WikToR dataset deteriorated so much after fine-tuning using any of the five training corpora. Furthermore, datasets such as CLDW or NCEN also showed unexpected results, with both positive and negative changes in performance, sometimes even varying between different subsets of the same training corpus. For example, fine-tuning with training data from GeoCorpora led to better overall performance on the CLDW dataset, except for the training subset of 400, for which performance dropped anomalously.

A possible explanation for these sudden changes in performance could be overfitting caused by specific training subsets, through which the model acquired a bias for certain interpretations of toponyms. For example, a sample from evaluations on the CLDW dataset showed that the baseline model correctly resolved 91% of the 533 occurrences of the toponym *'Penrith'*. After fine-tuning the model with increasing subsets from the GeoCorpora corpus, the rate of correctly resolved occurrences of this toponym fluctuated slightly but was particularly bad after fine-tuning with the subset of 400, after which the model was unable to resolve the toponym correctly even once. This suggests that the model learned to interpret this toponym in a specifically erroneous way based on new training examples introduced in that subset. Such rapidly developed distortions, especially for frequently occurring toponyms, would explain why the performance in some cases changed so drastically even from one subset to the next. The fact that such biases were able to be acquired with just small amounts of training data suggests issues with training hyperparameter settings, which were, in fact, not optimised in the context of these experiments.

7.2 Limitations

Although the experiments that were conducted provided valuable insights into the capabilities of Geoparser, there are a few limitations that affect the interpretation and generalisability of the results. A first limitation is presented by the fact that the experiments were conducted only with a GeoNames-based candidate generator, while the annotated toponyms were grounded with GeoNames for only six of the twelve datasets. Whether the source gazetteer used for annotations matches that of the candidate generator can, however, have a major influence on its ability to propose relevant candidates and, thus, also on the overall performance of toponym resolution. Generating candidates for toponyms in GeoNames-based datasets offers an inherent advantage for GeoNames-based candidate generators, as relevant candidates are guaranteed to exist in the gazetteer, as toponyms would otherwise not have been annotated with coordinates in the first place.

For toponyms in datasets that have not been grounded using GeoNames, however, it is likely that some locations will not be found in GeoNames. Toponyms may refer to locations that are missing from GeoNames, and thus, relevant candidates are potentially never proposed, preventing toponyms from being resolved correctly. Instead, they are grounded using other knowledge bases such as Wikipedia, OpenStreetMap or Unlock, which are likely to have been specifically selected because of more suitable coverages for the geographical scope of the respective datasets. Consequently, it is expected that a GeoNames-based candidate generator would perform worse on these datasets than on those based on GeoNames. However, since candidate generation recall could not be measured without annotated GeoNames IDs for toponyms, it remains unclear, for example, how much of the poor performance on the three historical corpora CLDW, NCEN and WOTR is ultimately attributable to the choice of gazetteer.

Next, the datasets used for the experiments are diverse in structure and thematic focus, allowing for comparisons to be made across different domains. Nevertheless, the datasets are still exclusively English-language texts. This substantially limits the generalisability of the findings and leaves open how well the proposed method might perform for other languages. That said, being able to use Geoparser for other languages is one of the fundamental design goals of the library. Users can choose from a variety of different language spaCy models, and there are several multilingual SentenceTransformer models that can be fine-tuned to disambiguate toponyms. However, it is unknown whether and how well the architecture of the Geoparser, which was designed based on requirements for English texts, will work for other languages. Different languages may have unique requirements for geoparsing tasks that extend beyond the language of neural models (Leppämäki et al., 2024). Investigating the applicability of Geoparser for individual languages will, therefore, be an important aspect of future work.

Replicating the evaluation framework of Hu et al. (2023a) for evaluating toponym resolution performance provided major benefits for comparisons with state-of-the-art systems. However, it also introduced limitations, especially with respect to the choice of evaluation metrics. Point-based error distance metrics such as A161, MED, and AUC can make it difficult to interpret system performances, which was also highlighted by Geoparser's metric-dependent performance ranking. Furthermore, the choice of gazetteers used for grounding locations can also considerably influence the computed performances (Leppämäki et al., 2024; Gritta et al., 2020). Different gazetteers may assign different coordinates to the same location, which can lead to erroneous interpretations of predictions during evaluation, especially for locations spanning large areas. This provides an additional advantage for GeoNames-based systems on GeoNames-grounded datasets, for which predictions of correct locations will always match the geographical positions provided in the annotations. Hu et al. (2023a) address this problem by removing toponyms of large-area locations such as *'Canada'* or *'Russia'* before the evaluation. A potentially better solution would be the use of polygon-based evaluation methods that assess predictions based on containment or overlap with annotated polygons (Leppämäki et al., 2024; Zhang & Bethard, 2024). However, such methods are difficult to implement due to the limited availability of polygon geometries in most gazetteers, especially for small-scale locations.

Finally, it should be mentioned that the experiments were conducted without any attempts to optimise performance. Many elements of the Geoparser configuration, such as the format of the textual representations of location candidates, the choice of the SentenceTransformer base model or the hyperparameters used for fine-tuning, were specified without investigating how different configurations could affect overall performance. This has, in some cases, led to issues, such as in the domain adaptation experiment, where poorly configured settings are likely the cause of the observed unexpected behaviours. Given the multitude of possible adjustments and settings that could be considered when configuring Geoparser, it is likely that a thorough optimisation of the individual components could potentially lead to improved performance. Therefore, an important goal for future work will be to systematically investigate and evaluate various configuration parameters to fully realise the potential of Geoparser.

7.3 Future Work

The presented results have demonstrated that Geoparser, in its current form, was able to achieve competitive performance and particularly stood out for its efficient use of a transformer model. However, it was outperformed by some state-of-the-art systems with respect to the accuracy of toponym resolution predictions and showed particularly poor performance on selected datasets. Given the still prototypical nature of Geoparser, it is likely that the full potential of the proposed method has not yet been realised and that the optimisation of individual system components could improve overall performance.

For example, there are a variety of different SentenceTransformer models that could be used as base models for creating embeddings. For the conducted experiments, the `all-distilroberta-v1` model was used, which was originally trained as a general-purpose model for text similarity tasks. However, other models exist that were trained for different specific tasks, such as semantic search or question answering, which could potentially have different effects on the final performance for the task of toponym resolution. There are also larger-scale SentenceTransformer models, which, although likely to have a negative impact on the efficiency of the system, could potentially lead to better performance. Finally, the SentenceTransformer framework also offers fundamentally different model architectures that may be better suited for disambiguating toponyms. One example would be an asymmetric model architecture, where two separate models would be employed for the encoding of candidates and toponyms.

Another important consideration when using SentenceTransformer models is the employed pooling method. In the current version, default mean pooling was used, which means that toponyms are represented based on averaged token embeddings of all the tokens in the provided context. For general text similarity tasks, this is a reasonable strategy to capture information about the entirety of the text. For toponym resolution, however, the focus should be on the toponym itself, which is not ideally done with mean pooling. Embeddings generated for toponyms are likely obscured by other words in the context, leading to potentially unreliable cues during disambiguation. A better alternative to mean pooling would be to directly use token embeddings of the toponyms in question. Token embeddings would likely represent toponyms more distinctively while still incorporating relevant information from the context, thanks to the attention mechanism of transformer models. However, implementing such a custom pooling method requires more complex adaptations to the SentenceTransformer framework, which natively does not support such representation formats.

Furthermore, default configurations provided by the SentenceTransformer library were used for the fine-tuning of models. In future work, these should be optimised for the task at hand to avoid potential bottlenecks caused by improper hyperparameters. Similarly, the choice of loss function used for training should also be investigated. For the conducted experiments, a contrastive loss function was used, which distinguishes between correct and incorrect candidates in a binary fashion. However, candidates may also be considered more or less similar for a particular toponym, and differentiating between different candidates, for example, based on their geographical position, could allow more nuanced distinctions between representations of locations. This could be implemented, for example, using a cosine similarity loss function, where distances of candidates to locations referenced by toponyms would be specified as continuous labels for training examples.

The construction of training examples also requires careful examination. Currently, a positive training example and a variable number of negative training examples are created for every toponym. In doing so, the number of negative training examples is limited only by the number of potential candidates for the respective toponym.

For unambiguous toponyms, on the other hand, no negative training examples are created at all since only the correct candidate is suggested for those. While this approach maximises the number of training examples that can be created from limited training corpora, it can also lead to largely unbalanced training data. The effect of this imbalance should be assessed and potentially addressed, for example, through random sampling of incorrect candidates and discarding training examples for unambiguous toponyms. Furthermore, other strategies for creating training examples could be explored. For example, a careful selection of candidates for creating negative training examples could potentially improve the quality of training data. This could be done, for example, based on the distances of candidates to toponym locations or by filtering candidates based on certain feature types.

Another aspect to be considered is the way in which candidate locations are textually represented. For the conducted experiments, candidates were represented using five attributes retrieved from GeoNames, which were used to construct an artificial sentence in the format: `[name] ([feature type]) in [admin2], [admin1], [country]`. Further attributes from GeoNames could potentially be incorporated, such as the name of third-order administrative divisions or numerical attributes, such as population numbers or geographical coordinates. Opportunities for integrating more detailed information about places would arise from using other knowledge bases like Wikidata, which contain more diverse information about places in the form of graph-based statements, such as `Zurich - part of - Greater Zurich Area`. Also, unstructured descriptions of locations, for example, in the form of text extracted from corresponding Wikipedia entries, as done by Wu et al. (2020) for BLINK, could be a way to increase information in location representations. However, choosing knowledge bases like these also means potentially limiting the coverage of toponyms, as they often generally have worse location coverage than specialised gazetteers like GeoNames.

Finally, an integral part of future work on the Geoparser library will be the integration of additional gazetteers or knowledge bases for candidate generation. It was illustrated how much the quality of toponym resolution results depends on the employed gazetteer being suitable for the requirements of individual text corpora. These requirements may be geographic, like the availability of fine-grained locations for specific regions, or thematic, such as coverage of historical toponyms. Extending the library to include additional information sources is an important part of the ability to adapt the geoparsing pipeline to individual requirements. The architecture of Geoparser provides the necessary interface for such extensions, which makes it possible to integrate new gazetteers in a modular way.

8 Conclusion

This thesis has explored the use of transformer models for toponym resolution. The motivation behind this work was the challenge to make use of the advanced natural language processing capabilities of transformer models for disambiguating location references without compromising efficiency and scalability. To this end, a new method was proposed that uses a bi-encoder-based ranking approach built on the SentenceTransformers framework, allowing toponyms within texts to be efficiently compared with location candidates from a gazetteer.

One of the main contributions of this work was integrating the proposed method into a dedicated geoparsing pipeline, which was published as an open-source Python library under the name Geoparser. The library makes use of the NER functionality of spaCy for toponym recognition and uses specially fine-tuned SentenceTransformer models for disambiguating toponyms. Thanks to its modular architecture, the library can be flexibly adapted to different scenarios. Users can employ different models and knowledge bases to suit individual requirements of specific text corpora and customise models by fine-tuning them or training them from scratch.

Geoparser was evaluated by replicating the evaluation framework of Hu et al. (2023a), which allowed direct comparability with existing toponym resolution systems. The experiments showed that the proposed method achieved competitive performance on several datasets, standing out in particular for its low runtimes. The system's ability to adapt to different text domains was also investigated. By further fine-tuning the model on small amounts of data from domain-specific corpora, improvements in performance on corresponding datasets were obtained.

In summary, this work has shown that transformer models can indeed be used both efficiently and effectively for resolving toponyms despite their computationally expensive nature. Given the prototypical nature of Geoparser, it is likely that optimisations to the model architecture and training processes could further improve the overall performance of the system. Given the flexibility of Geoparser to be easily adapted for different scenarios, it has the potential to become a viable solution for a wide range of real-world applications.

References

Adams, Benjamin, Grant McKenzie & Mark Gahegan. 2015. Frankenplace: Interactive Thematic Mapping for Ad Hoc Exploratory Search. In *Proceedings of the 24th International Conference on World Wide Web* WWW '15, 12–22. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2736277.2741137.

Amitay, Einat, Nadav Har'El, Ron Sivan & Aya Soffer. 2004. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* SIGIR '04, 273–280. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1008992.1009040.

Ardanuy, Mariona Coll, David Beavan, Kaspar Beelen, Kasra Hosseini, Jon Lawrence, Katherine McDonough, Federico Nanni, Daniel van Strien & Daniel C. S. Wilson. 2022. A Dataset for Toponym Resolution in Nineteenth-Century English Newspapers. *Journal of Open Humanities Data* 8(0). https://doi.org/10.5334/johd.56.

Berico Technologies. 2012. Cartographic Location and Vicinity Indexer (CLAVIN).

Buscaldi, Davide. 2011. Approaches to disambiguating toponyms. *SIGSPATIAL Special* 3(2). 16–19. https://doi.org/10.1145/2047296.2047300.

Cai, Guoray. 2002. GeoVSM: An Integrated Retrieval Model for Geographic Information. In Max J. Egenhofer & David M. Mark (eds.), *Geographic Information Science*, 65–79. Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-45799-2_5.

Cardoso, Ana Bárbara, Bruno Martins & Jacinto Estima. 2022. A Novel Deep Learning Approach Using Contextual Embeddings for Toponym Resolution. *ISPRS International Journal of Geo-Information* 11(1). 28. https://doi.org/10.3390/ijgi11010028.

De Cao, Nicola, Gautier Izacard, Sebastian Riedel & Fabio Petroni. 2021. Autoregressive Entity Retrieval. ArXiv:2010.00904 [cs, stat]. https://doi.org/10.48550/arXiv.2010.00904.

DeLozier, Grant, Jason Baldridge & Loretta London. 2015. Gazetteer-Independent Toponym Resolution Using Geographic Word Profiles. *Proceedings of the AAAI Conference on Artificial Intelligence* 29(1). https://doi.org/10.1609/aaai.v29i1.9531.

DeLozier, Grant, Ben Wing, Jason Baldridge & Scott Nesbit. 2016. Creating a Novel Geolocation Corpus from Historical Texts. In Annemarie Friedrich & Katrin Tomanek (eds.), *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, 188–198. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-1721.

Derungs, Curdin & Ross S. Purves. 2016. Mining nearness relations from an n-grams Web corpus in geographical space. *Spatial Cognition & Computation* 16(4). 301–322. https://doi.org/10.1080/13875868.2016.1246553.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran & Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423.

Goodchild, M. F. & L. L. Hill. 2008. Introduction to digital gazetteer research. *International Journal of Geographical Information Science* 22(10). 1039–1044. https://doi.org/10.1080/13658810701850497.

Gritta, Milan, Mohammad Taher Pilehvar & Nigel Collier. 2018a. Which Melbourne? Augmenting Geocoding with Maps. In Iryna Gurevych & Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1285–1296. Melbourne, Australia: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1119.

Gritta, Milan, Mohammad Taher Pilehvar & Nigel Collier. 2020. A pragmatic guide to geoparsing evaluation. *Language Resources and Evaluation* 54(3). 683–712. https://doi.org/10.1007/s10579-019-09475-3.

Gritta, Milan, Mohammad Taher Pilehvar, Nut Limsopatham & Nigel Collier. 2018b. What's missing in geographical parsing? *Language Resources and Evaluation* 52(2). 603–623. `https://doi.org/10.1007/s10579-017-9385-8`.

Grover, Claire & Richard Tobin. 2014. A Gazetteer and Georeferencing for Historical English Documents. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, 119–127. Gothenburg, Sweden: Association for Computational Linguistics. `https://doi.org/10.3115/v1/W14-0617`.

Halterman, Andrew. 2023. Mordecai 3: A Neural Geoparser and Event Geocoder. ArXiv:2303.13675 [cs]. `https://doi.org/10.48550/arXiv.2303.13675`.

Harris, Zellig S. 1954. Distributional Structure. *WORD* 10(2-3). 146–162. `https://doi.org/10.1080/00437956.1954.11659520`.

Hosseini, Kasra, Federico Nanni & Mariona Coll Ardanuy. 2020. DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching. In Qun Liu & David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 62–69. Online: Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.emnlp-demos.9`.

Hu, Xuke & Jens Kersten. 2024. Enhancing Toponym Resolution with Fine-Tuned LLMs (Llama2). In Xuke Hu, Ross Purves, Ludovic Moncla, Jens Kersten & Kristin Stock (eds.), *Proceedings of The GeoExT 2024: Geographic Information Extraction from Texts Workshop*, vol. 3683 CEUR Workshop Proceedings, 52–56. Glasgow, Scotland: CEUR. `https://ceur-ws.org/Vol-3683/paper7.pdf`.

Hu, Xuke, Yeran Sun, Jens Kersten, Zhiyong Zhou, Friederike Klan & Hongchao Fan. 2023a. How can voting mechanisms improve the robustness and generalizability of toponym disambiguation? *International Journal of Applied Earth Observation and Geoinformation* 117. 103191. `https://doi.org/10.1016/j.jag.2023.103191`.

Hu, Xuke, Zhiyong Zhou, Hao Li, Yingjie Hu, Fuqiang Gu, Jens Kersten, Hongchao Fan & Friederike Klan. 2023b. Location Reference Recognition from Texts: A Survey and Comparison. *ACM Comput. Surv.* 56(5). 112:1–112:37. `https://doi.org/10.1145/3625819`.

Jones, Christopher B. & Ross S. Purves. 2008. Geographical information retrieval. *International Journal of Geographical Information Science* 22(3). 219–228. `https://doi.org/10.1080/13658810701626343`.

Jurafsky, Daniel & James H. Martin. 2024. *Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing*. Upper Saddle River, NJ: Prentice Hall. https://web.stanford.edu/~jurafsky/slp3/.

Kamalloo, Ehsan & Davood Rafiei. 2018. A Coherent Unsupervised Model for Toponym Resolution. In *Proceedings of the 2018 World Wide Web Conference* WWW '18, 1287–1296. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3178876.3186027.

Karimzadeh, Morteza & Alan M. MacEachren. 2019. GeoAnnotator: A Collaborative Semi-Automatic Platform for Constructing Geo-Annotated Text Corpora. *ISPRS International Journal of Geo-Information* 8(4). 161. https://doi.org/10.3390/ijgi8040161.

Katz, Philipp & Alexander Schill. 2013. To Learn or to Rule: Two Approaches for Extracting Geographical Information from Unstructured Text. In *Proceedings of the 11-th Australasian Data Mining Conference (AusDM'13)*, vol. 146, Canberra, Australia. https://crpit.scem.westernsydney.edu.au/abstracts/CRPITV146Katz.

Krovetz, Robert & W. Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst.* 10(2). 115–141. https://doi.org/10.1145/146802.146810.

Larson, Ray R. 1996. Geographic information retrieval and spatial browsing. *Geographic information systems and libraries: patrons, maps, and spatial information [papers presented at the 1995 Clinic on Library Applications of Data Processing, April 10-12, 1995]* https://hdl.handle.net/2142/416.

Larson, Ray R. & Patricia Frontiera. 2004. Spatial Ranking Methods for Geographic Information Retrieval (GIR) in Digital Libraries. In Rachel Heery & Liz Lyon (eds.), *Research and Advanced Technology for Digital Libraries*, 45–56. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-30230-8_5.

Leidner, Jochen L. 2007. Toponym resolution in text: annotation, evaluation and applications of spatial grounding. *ACM SIGIR Forum* 41(2). 124–126. https://doi.org/10.1145/1328964.1328989.

Leidner, Jochen L. & Michael D. Lieberman. 2011. Detecting geographical references in the form of place names and associated spatial natural language.

SIGSPATIAL Special 3(2). 5–11. https://doi.org/10.1145/2047296.2047298.

Leppämäki, Tatu, Tuuli Toivonen & Tuomo Hiippala. 2024. Geographical and linguistic perspectives on developing geoparsers with generic resources. *International Journal of Geographical Information Science* 0(0). 1–22. https://doi.org/10.1080/13658816.2024.2369539.

Leveling, Johannes & Sven Hartrumpf. 2008. On metonymy recognition for geographic information retrieval. *International Journal of Geographical Information Science* 22(3). 289–299. https://doi.org/10.1080/13658810701626244.

Li, Zekun, Wenxuan Zhou, Yao-Yi Chiang & Muhao Chen. 2023. GeoLM: Empowering Language Models for Geospatially Grounded Language Understanding. ArXiv:2310.14478 [cs]. https://doi.org/10.48550/arXiv.2310.14478.

Lieberman, Michael D. & Hanan Samet. 2012. Adaptive context features for toponym resolution in streaming news. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* SIGIR '12, 731–740. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2348283.2348381.

Lieberman, Michael D., Hanan Samet & Jagan Sankaranarayanan. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 201–212. https://doi.org/10.1109/ICDE.2010.5447903.

Liu, Fangyu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella & Nigel Collier. 2021. Self-Alignment Pretraining for Biomedical Entity Representations. ArXiv:2010.11784 [cs]. https://doi.org/10.48550/arXiv.2010.11784.

Louwerse, Max M. & Rolf A. Zwaan. 2009. Language Encodes Geographical Information. *Cognitive Science* 33(1). 51–73. https://doi.org/10.1111/j.1551-6709.2008.01003.x.

Machado, Ivre Marjorie R., Rafael Odon de Alencar, Roberto de Oliveira Campos & Clodoveu A. Davis. 2011. An ontological gazetteer and its application for place name disambiguation in text. *Journal of the Brazilian Computer Society* 17(4). 267–279. https://doi.org/10.1007/s13173-011-0044-4.

Manning, Christopher D., Prabhakar Raghavan & Hinrich Schütze. 2008. *Introduction to information retrieval.* New York: Cambridge University Press. https://nlp.stanford.edu/IR-book/information-retrieval-book.html.

Mikolov, Tomas, Kai Chen, Greg Corrado & Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. ArXiv:1301.3781 [cs]. https://doi.org/10.48550/arXiv.1301.3781.

Nadeau, David & Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticæ Investigationes* 30(1). 3–26. https://doi.org/10.1075/li.30.1.03nad.

Orr, Laurel, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling & Christopher Re. 2020. Bootleg: Chasing the Tail with Self-Supervised Named Entity Disambiguation. ArXiv:2010.10363 [cs]. https://doi.org/10.48550/arXiv.2010.10363.

Overell, Simon. 2011. The problem of place name ambiguity. *SIGSPATIAL Special* 3(2). 12–15. https://doi.org/10.1145/2047296.2047299.

Purves, Ross S., Paul Clough, Christopher B. Jones, Avi Arampatzis, Benedicte Bucher, David Finch, Gaihua Fu, Hideo Joho, Awase Khirni Syed, Subodh Vaid & Bisheng Yang. 2007. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet. *International Journal of Geographical Information Science* 21(7). 717–745. https://doi.org/10.1080/13658810601169840.

Purves, Ross S., Paul Clough, Christopher B. Jones, Mark H. Hall & Vanessa Murdock. 2018. Geographic Information Retrieval: Progress and Challenges in Spatial Search of Text. *Foundations and Trends® in Information Retrieval* 12(2-3). 164–318. https://doi.org/10.1561/1500000034.

Radford, Alec, Karthik Narasimhan, Tim Salimans & Ilya Sutskever. 2018. Improving language understanding by generative pre-training https://www.mikecaptain.com/resources/pdf/GPT-1.pdf. Publisher: San Francisco, CA, USA.

Radford, Benjamin J. 2021. Regressing Location on Text for Probabilistic Geocoding. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, 53–57. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.case-1.8.

Rayson, Paul, Alex Reinhold, James Butler, Chris Donaldson, Ian Gregory & Joanna Taylor. 2017. A deeply annotated testbed for geographical text analysis: The Corpus of Lake District Writing. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities* GeoHumanities '17, 9–15. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3149858.3149865.

Reimers, Nils & Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410.

Rizzo, Giuseppe & Marieke van Erp. 2016. Making Sense of Microposts (#Microposts2016) Named Entity rEcognition and Linking (NEEL) Challenge https://ceur-ws.org/Vol-1691/microposts2016_neel-challenge-report.

Schockaert, Steven & Martine De Cock. 2007. Reasoning about vague topological information. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* CIKM '07, 593–602. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1321440.1321524.

Solaz, Yuval & Vitaly Shalumov. 2023. Transformer Based Geocoding. ArXiv:2301.01170 [cs]. https://doi.org/10.48550/arXiv.2301.01170.

Speriosu, Michael & Jason Baldridge. 2013. Text-Driven Toponym Resolution using Indirect Supervision. In Hinrich Schuetze, Pascale Fung & Massimo Poesio (eds.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1466–1476. Sofia, Bulgaria: Association for Computational Linguistics. https://aclanthology.org/P13-1144.

Vasardani, Maria, Stephan Winter & Kai-Florian Richter. 2013. Locating place names from place descriptions. *International Journal of Geographical Information Science* 27(12). 2509–2532. https://doi.org/10.1080/13658816.2013.785550.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. 2017. Attention Is All You Need. ArXiv:1706.03762 [cs]. https://doi.org/10.48550/arXiv.1706.03762.

Wallgrün, Jan Oliver, Morteza Karimzadeh, Alan M. MacEachren & Scott Pezanowski. 2018. GeoCorpora: building a corpus to test and train microblog geoparsers. *International Journal of Geographical Information Science* 32(1). 1–29. https://doi.org/10.1080/13658816.2017.1368523.

Weissenbacher, Davy, Arjun Magge, Karen O'Connor, Matthew Scotch & Graciela Gonzalez-Hernandez. 2019. SemEval-2019 Task 12: Toponym Resolution in Scientific Papers. In Jonathan May, Ekaterina Shutova, Aurelie Herbelot, Xiaodan Zhu, Marianna Apidianaki & Saif M. Mohammad (eds.), *Proceedings of the 13th International Workshop on Semantic Evaluation*, 907–916. Minneapolis, Minnesota, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/S19-2155.

Wu, Ledell, Fabio Petroni, Martin Josifoski, Sebastian Riedel & Luke Zettlemoyer. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In Bonnie Webber, Trevor Cohn, Yulan He & Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6397–6407. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.519.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes & Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. ArXiv:1609.08144 [cs]. https://doi.org/10.48550/arXiv.1609.08144.

Xie, Tingyu, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu & Hongwei Wang. 2023. Empirical Study of Zero-Shot NER with ChatGPT. ArXiv:2310.10035 [cs]. https://doi.org/10.48550/arXiv.2310.10035.

Yang, Xiyuan, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu & Xiang Ren. 2019. Learning Dynamic Context Augmentation for Global Entity Linking. In Kentaro Inui, Jing Jiang, Vincent Ng & Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 271–281. Hong

Kong, China: Association for Computational Linguistics. `https://doi.org/10.18653/v1/D19-1026`.

Zhang, Zeyu & Steven Bethard. 2023. Improving Toponym Resolution with Better Candidate Generation, Transformer-based Reranking, and Two-Stage Resolution. ArXiv:2305.11315 [cs]. `https://doi.org/10.48550/arXiv.2305.11315`.

Zhang, Zeyu & Steven Bethard. 2024. A survey on geocoding: algorithms and datasets for toponym resolution. *Language Resources and Evaluation* `https://doi.org/10.1007/s10579-024-09730-2`.

Zhang, Zeyu, Egoitz Laparra & Steven Bethard. 2024. Improving Toponym Resolution by Predicting Attributes to Constrain Geographical Ontology Entries. In Kevin Duh, Helena Gomez & Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 35–44. Mexico City, Mexico: Association for Computational Linguistics. `https://doi.org/10.18653/v1/2024.naacl-short.3`.

Appendix

The following pages contain supplementary figures referenced in Chapter 6, providing additional visualisations of the results in terms of Mean Error Distance and Area Under the Curve.

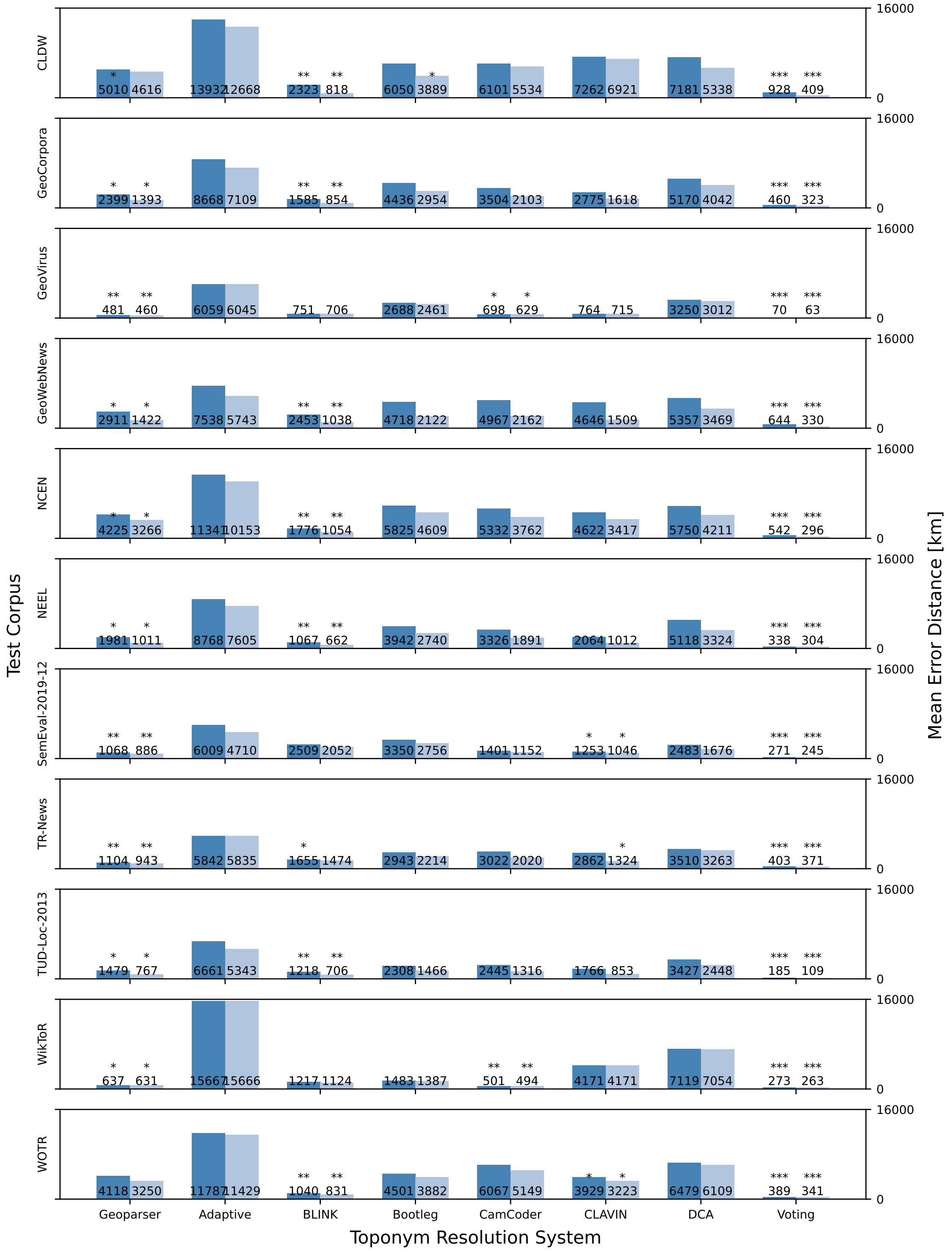

Figure 5: Toponym resolution Mean Error Distance on all toponyms in the datasets (dark blue) and on the spaCy subsets (light blue); *** indicates the best system, ** the second-best, and * the third-best

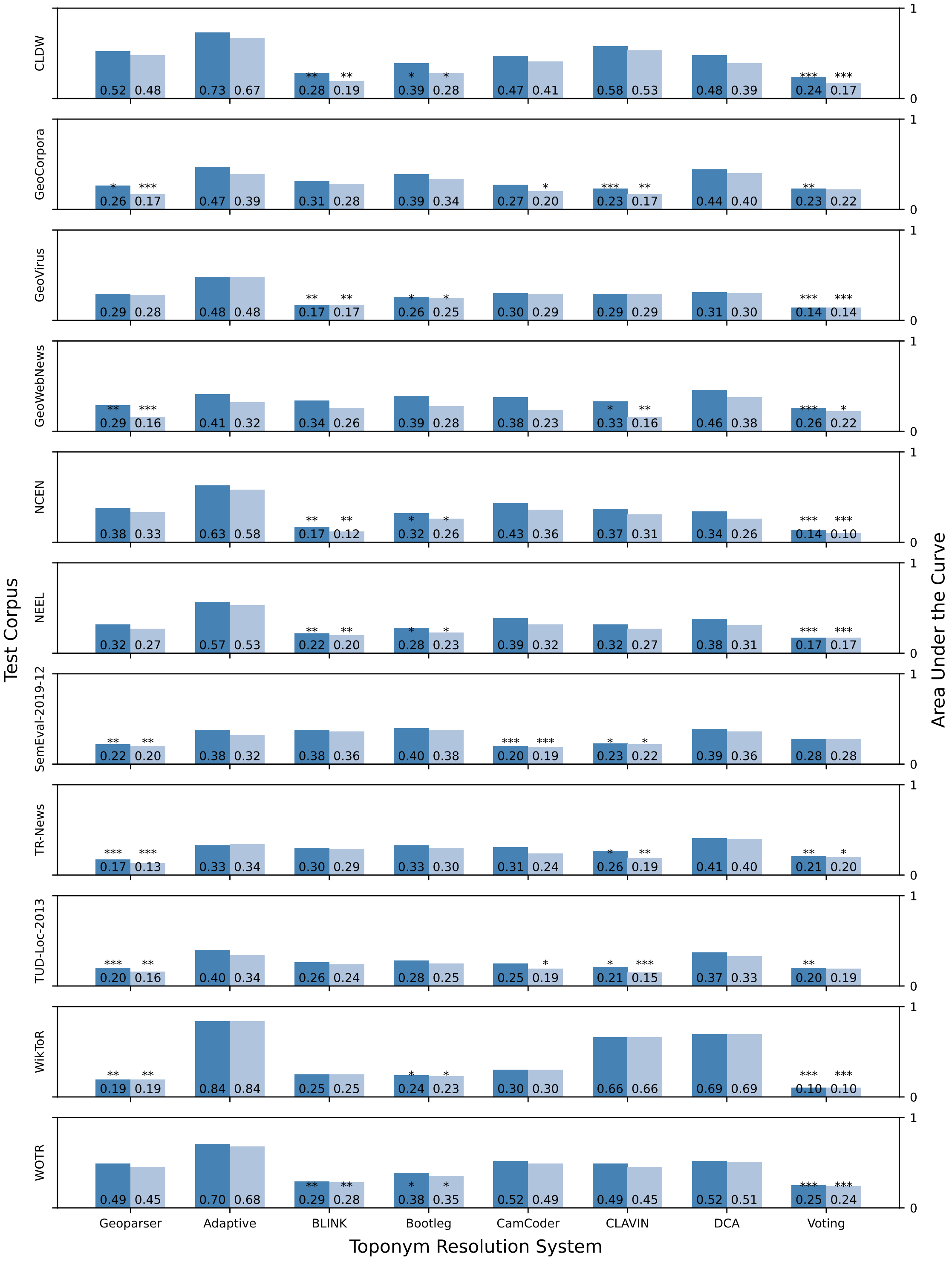

Figure 6: Toponym resolution Area Under the Curve on all toponyms in the datasets (dark blue) and on the spaCy subsets (light blue); *** indicates the best system, ** the second-best, and * the third-best

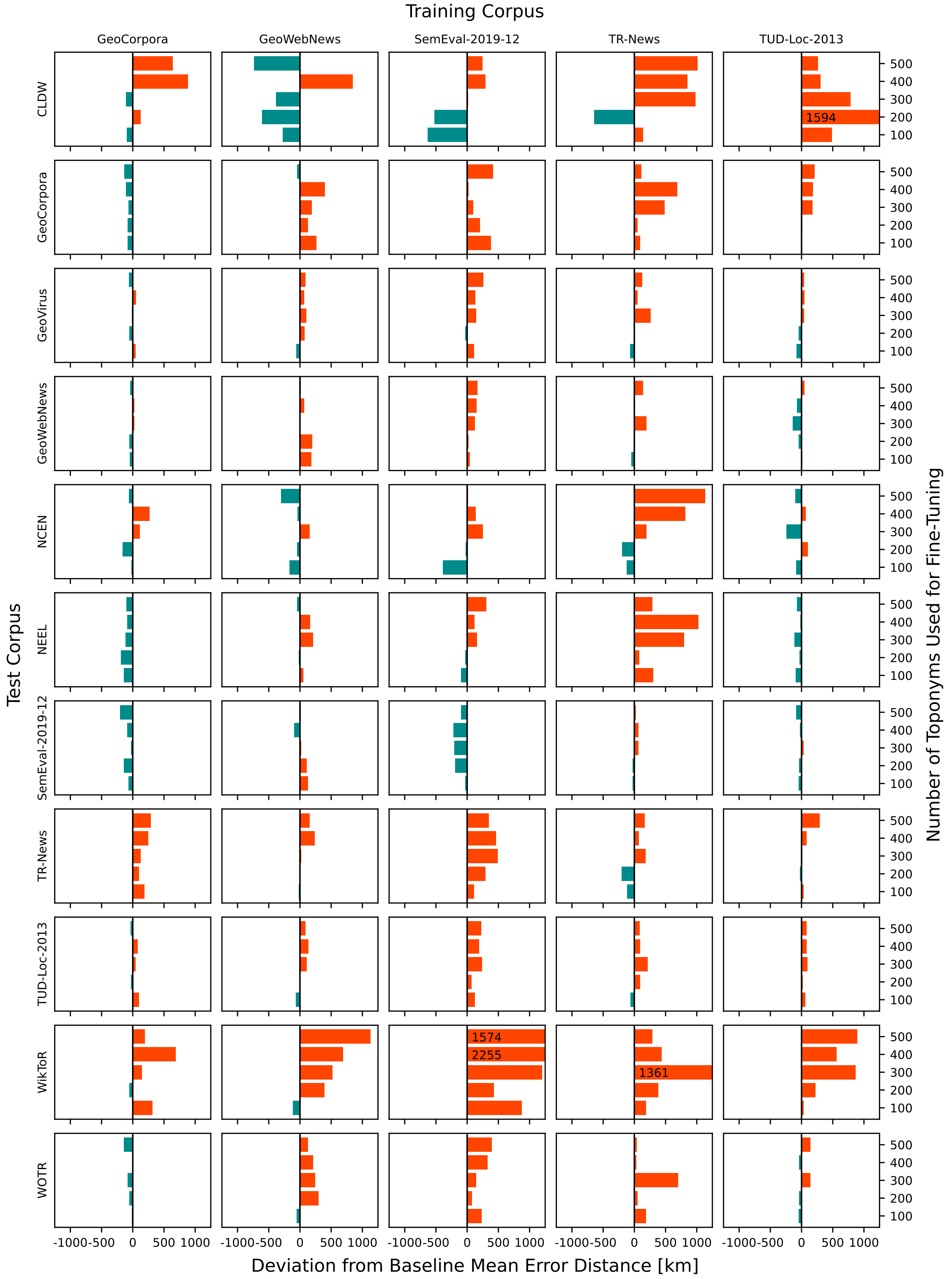

Figure 7: Mean Error Distance deviation from baseline model after further fine-tuning (model deterioration in orange and improvement in green)

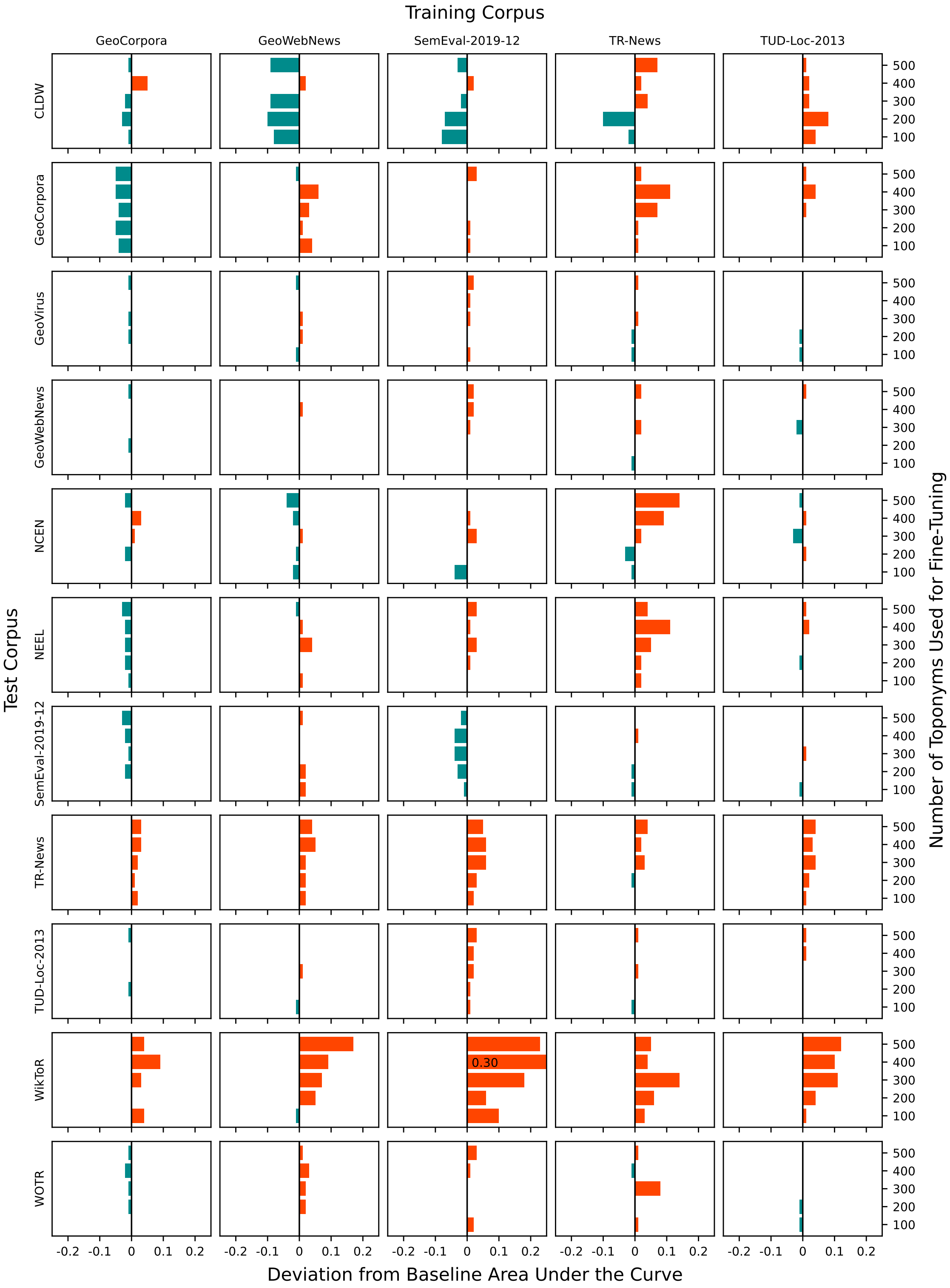

Figure 8: Area Under the Curve deviation from baseline model after further fine-tuning (model deterioration in orange and improvement in green)

Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

I further declare that I used ChatGPT, DeepL, and Grammarly for rephrasing, translation, and grammar and style corrections. Nonetheless, I assume full responsibility for the content of this thesis.

D.Gomes

Diego Gomes, 30.09.2024