

Article

# GeoAnnotator: A Collaborative Semi-Automatic Platform for Constructing Geo-Annotated Text Corpora

Morteza Karimzadeh <sup>1,\*</sup>  and Alan M. MacEachren <sup>2</sup> 

<sup>1</sup> School of Electrical and Computer Engineering, Purdue University, IN 47907, USA

<sup>2</sup> Department of Geography, Penn State, PA 16802, USA; maceachren@psu.edu

\* Correspondence: karimzadeh@purdue.edu; Tel.: +1-765-494-2273

Received: 29 November 2018; Accepted: 15 March 2019; Published: 27 March 2019



**Abstract:** Ground-truth datasets are essential for the training and evaluation of any automated algorithm. As such, gold-standard annotated corpora underlie most advances in natural language processing (NLP). However, only a few relatively small (geo-)annotated datasets are available for geoparsing, i.e., the automatic recognition and geolocation of place references in unstructured text. The creation of geoparsing corpora that include both the recognition of place names in text and matching of those names to toponyms in a geographic gazetteer (a process we call geo-annotation), is a laborious, time-consuming and expensive task. The field lacks efficient geo-annotation tools to support corpus building and lacks design guidelines for the development of such tools. Here, we present the iterative design of GeoAnnotator, a web-based, semi-automatic and collaborative visual analytics platform for geo-annotation. GeoAnnotator facilitates collaborative, multi-annotator creation of large corpora of geo-annotated text by generating computationally-generated pre-annotations that can be improved by human-annotator users. The resulting corpora can be used in improving and benchmarking geoparsing algorithms as well as various other spatial language-related methods. Further, the iterative design process and the resulting design decisions can be used in annotation platforms tailored for other application domains of NLP.

**Keywords:** geoparsing; iterative design; design guidelines; annotation; corpus; spatial linguistics; geographic information retrieval

## 1. Introduction

Geoparsing is defined as the process of automatically resolving place references in natural language (unstructured text) to toponyms in a geographic gazetteer with geographic coordinates. Geoparsing enables the extraction of textual information about places for use in geographic information systems (GIS) and other applications. For instance, geoparsing the sentence “the Paradise Papers showed 1,000,000 GBP of the Queen’s private money was invested offshore in Bermuda and the Cayman Islands”, would result in the recognition (detection) of “Bermuda” and “Cayman Islands” as place names, assigning the toponyms of [Bermuda, GeoNames (geonames.org) ID of 65365, coordinates of N 32°19′49″ W 64°44′24″] and [Cayman Islands, GeoNames ID of 3580718, coordinates of N 19°30′00″ W 80°40′00″] to the recognized named entities (NE), respectively. Linguistically annotated corpora underlie most advances in natural language processing (NLP) [1]. While many annotated corpora exist in various domains for evaluating and training statistical NLP methods, there are only a few publically available corpora available for geoparsing [2–5]. Evaluating geoparsing using ground-truth corpora is a key step in improving performance and proposing new solutions. The lack of available manually annotated gold-standard corpora is an impediment to devising performant geoparsing algorithms and

advancing the field of geographic information retrieval (GIR) [3,4,6]. To create unbiased corpora that capture the complexity of natural language and place name ambiguity and annotate with ground-truth toponyms from high coverage and detailed gazetteers, documents used for testing and training should be manually “geo-annotated” by human annotators, i.e., place names should be recognized (segmented) and manually resolved to toponyms in gazetteers. The process of manually tagging (segmentation) and annotating place names in text with entries (toponyms) from a geographic gazetteer, here called “Geo-Annotation”, is laborious, costly, time-consuming and error-prone. The scarcity of publically available geo-annotated corpora can partially be attributed to the lack of available efficient software infrastructure capable of facilitating this laborious task. Annotation is a challenge in its own right that has attracted research, particularly from the linguistics community. Having an efficient and reliable annotation tool is central to being able to build large gold-standard datasets that can be useful for training computational classifiers.

In this article, we present GeoAnnotator (source code available at <https://github.com/geovista/GeoTxt>): A web-based, self-sufficient, semi-automatic and collaborative visual analytics platform for geo-annotation. GeoAnnotator facilitates the difficult process of corpus building by providing the initial geo-annotation of text (pre-geo-annotation), an interactive linked map and text views for maximum efficiency, (geo-)annotation agreement evaluation functions for collaborative corpus building and built-in semi-automatic consistency-checking functionality that supports human annotators in the process of refining the pre-geo-annotation.

GeoAnnotator enables the iterative and coordinated annotation of documents. For each document, the final product includes identified place names, their position in the text (character offset) and the gazetteer toponym(s) to which each name is resolved. GeoAnnotator uses GeoNames as its default gazetteer because of its frequent updates, ease of ability by users to correct information, extensive coverage and quality [7] and inclusion of metadata items necessary for geoparsing/geo-annotation, such as alternative names and spatial hierarchies. All GeoNames toponyms have a unique identifier, enabling creation of corpora useful for linked data applications.

Our major contributions include: (1) Documentation of the iterative design process followed to develop an efficient visual analytics application (GeoAnnotator) to support the process of geo-annotation, (2) a detailed description of the functionality of this application and the reasoning behind design decisions made in its development, and (3) insights about the challenges of geo-annotation gained by developing and using GeoAnnotator to produce a corpus of tweets posted to Twitter with tagged locations. Through the iterative design process, we identified sources of inefficiency and disagreement in geo-annotation and special cases for place names and toponyms (such as overlapping or ambiguous place names), and iteratively improved the usefulness and usability of GeoAnnotator to better support geo-annotation. Such special cases should also be addressed by automatic geoparsers, therefore, the special cases and design decisions that we present in this article have implications for geoparsing and spatial language studies as well. Additionally, the concepts and techniques introduced here can be adopted in designing annotation platforms for other kinds of entities (e.g., persons or organizations).

## 2. Background and Related Work

Not surprisingly, many software applications have been developed to help support the process of building corpora of all kinds. We provide a brief review of annotation tools in other domains (tools that focus on annotating text for features beyond place names, such as persons, organizations, times, and more), and then focus on geo-annotation tools tailored to geoparsing and spatial language studies.

### 2.1. Review of Annotation Tools in Other Domains

In 2013, Bontcheva et al. reviewed nine general text annotation tools (primarily tools for tagging words or phrases by the type of feature signified), systematically comparing the key capabilities of the tools [8]. Most of the tools summarized in that review are implemented as desktop applications,

meaning that they require software installation on the client side. Those implemented as web tools include Atlas.ti (<http://www.atlasti.com/>), some tools in the Linguistic Data Consortium (LDC) toolset (<https://www.ldc.upenn.edu/language-resources/tools>), OntoNotes [9] and GATE Teamware [8]. An additional desktop tool not included in that review by Bontcheva et al. (2013) is XCon Suite, used in the annotation of the GENIA dataset [10], which deals with biological events [11].

Recent annotation platforms are increasingly being implemented with web user interfaces (UIs) that eliminate the need for users to install specialized software on their computers. These web-based tools enable the creation of larger corpora by crowdsourcing the annotation task to workers hired through platforms such as Amazon Mechanical Turk (<https://www.mturk.com/mturk/welcome>) or Figure Eight (<https://www.figure-eight.com/>). Sabou et al. (2014) review 13 such research studies, leveraging crowdsourcing for the annotation of large corpora, also proposing best practice guidelines for crowdsourcing annotation [12].

There are several annotation tools with web UIs. These include BioNotate, which is used in biomedical research for the annotation of binary relations such as protein-protein (association) and gene-disease (interaction) [13]; Phrase Detectives 2, a game-based tool to collect judgments about anaphoric annotations [14]; and Djangology, a light-weight web annotation platform for the distributed annotation of named entities that includes an administrator dashboard displaying inter-annotator agreement statistics for the generated corpus [15]. Another tool, named AlvisAE, used in domains of biology and crop science, introduces two important innovations: (1) The capability to assign different roles and annotation tasks to different annotators based on their level of expertise, and (2) automatically generated pre-annotations, which users can modify [16]. Lastly, BRAT is a web-based annotation tool that has been widely used to support a variety of natural language processing (NLP) tasks such as part-of-speech tagging, named entity recognition (NER) or the connectivity of annotations based on binary relations [17]. In an experimental annotation of biomolecular science articles, Stenetorp et al. (2012) reported a 15% decrease in annotation time when automatically generated pre-annotations were suggested to users [17].

GATE Teamware is an open-source, web accessible, collaborative annotation platform that allows for different user roles such as annotator, manager, and administrator [8]. GATE Teamware also has pre-annotation capabilities. Meurs et al. (2011) report that using GATE Teamware for the automatic pre-annotation and manual correction of fungal enzymes for use in biofuel research increased the annotation speed by about 50% when compared to fully manual annotation [18].

Annotation applications fall into two general categories in terms of the generated annotation output artifact: (1) Those that support named entity (NE) annotation (i.e., text segmentation) and the assignment of entity types to text segments/spans (such as toponym recognition, e.g., annotating Washington as an entity of “place”-type in the phrase “I live in Washington”); and (2) those that also support the disambiguation of identified names or phrases to unique entries in a knowledge base (such as resolving the place name of Washington to the toponym of “Washington, D.C.” in the GeoNames knowledge base). All annotation tools named so far under this section fall under the first category.

Bontcheva et al. (2014) conducted two case studies in the crowdsourcing of annotations of both named entities and entity disambiguation to the DBpedia knowledge base, and used the GATE crowdsourcing plugin to transfer documents to the crowdsourcing website CrowdFlower (<https://www.crowdflower.com/>) and back for NLP classification [12,19]. NE annotation and resolving to DBpedia were performed in two separate stages using two separate interfaces. Simple text highlighting was used for NE annotation. The resulting annotated NEs were used in a separate user interface, in which for each NE, a list of potential candidates (including place candidates) from DBpedia was presented to the users, who then chose the best candidate from the list.

None of the above tools support resolving place names to GeoNames, the richest gazetteer available at the time of this writing [7]. Further, geo-annotation with toponyms requires at least (1) an interactive map interface (linked with text annotation), (2) strategies for managing geo-annotation provided by multiple annotators, and (3) the ability of software to assign more than one toponym to a

place name (necessary for cases when more than one correct toponym can be identified by humans). While it may have been possible to modify and extend one of the existing text annotation platforms to support efficient geo-annotation, the fact that none were explicitly built with that goal in mind prompted us to design the system presented here from the ground up. In doing so, we drew upon key conceptual ideas from some of the systems just described as well as from some existing geo-annotation tools which are reviewed in the next subsection.

## 2.2. Geo-Annotation Tools

There are multiple corpora with manually annotated NE (including place NE) available to support NER research (such as [20–22]). These corpora can be used for evaluating and training toponym recognition methods. However, much less attention has been paid to constructing geo-annotated corpora (with place names resolved to toponyms in a gazetteer) [2–4,23].

Due to the shortage of available geo-annotated corpora, some researchers have generated automatically harvested corpora from existing sources on the web. For instance, Gritta et al. (2017) harvested a corpus by creating a list of the most ambiguous names on GeoNames, removing duplicates of those names, then retrieving Wikipedia entries for the remaining names in the list [4]. The Wikipedia page's first paragraph constitutes an artifact in the corpus, with the place name from the initial list being the only one that is geotagged. For instance, a page about "Springfield" is annotated with only the occurrences of "Springfield" and its geographic location. This approach also ignores the case for the existence of ambiguous place names that overlap one another (e.g., "Guam" which is used to indicate the island of Guam as well as the dependent political entity located on the island, or "Lagos" which refers to the seat of a second-order administrative division as well as a first-order administrative division). The ability to distinguish between different geographic feature types or prominence levels is especially important for linked data approaches that depend on a unique identifier for each entity. Ju et al. (2016) presented another similarly automatically harvested corpus of 5500 sentences, in which ambiguous place names were extracted from a Wikipedia list ([https://en.wikipedia.org/wiki/List\\_of\\_the\\_most\\_common\\_U.S.\\_place\\_names](https://en.wikipedia.org/wiki/List_of_the_most_common_U.S._place_names)) of the most common U.S. place names [24]. The qualifying names from that list (such as Springfield, MI) were then used to retrieve webpages containing the qualifying name, and then the qualifier of the place name was removed (MI is removed from Springfield, MI), so that the remaining place name was ambiguous. The only geotagged place name in the retrieved sentence is the one that originated the query. Therefore, it would be impossible to measure the performance of a geoparsing algorithm trained by (or designed for) such a corpus that includes multiple place names. While such automatically harvested corpora can be useful in creating geoparsers that only address highly ambiguous names, a corpus that is biased towards a selection of expected errors would be likely to have a negative impact on training geoparsers for text that includes other kinds of problems (due to overfitting). Moreover, using existing web pages on the internet to harvest geo-annotated corpora may aggravate the digital divide, because different places and countries that are already linked to geographic coordinates are not represented on the World Wide Web and geographic gazetteers equally [25].

For a geoparsing evaluation to be unbiased in terms of: (1) Spatial coverage (e.g., places in different continents) and types of places (e.g., populated places versus physical features), (2) time periods and events mentioned in text, and (3) linguistic diversity (e.g., place informal names and nicknames), the careful sampling of documents and manual geo-annotation is required. Such manual geo-annotation requires a well-designed annotation platform for accurate and efficient annotation, which is the primary focus of the research we report here.

In an early attempt to create such an annotation tool, Leidner (2007) developed Toponym Annotation Markup Editor (TAME), which is probably the first textual geo-annotation software [5]. TAME allowed human annotators to manually resolve previously identified place names in text to toponyms in the GEPmet Names Server (GNS) gazetteers (of the National Geospatial Intelligence Agency) and the Geographic Names Information System (GNIS) gazetteer (of U.S. Geological Survey),

both of which subsequently created the backbone of the GeoNames gazetteer. Implemented as a web page, TAME did not offer a map interface and used dropdown boxes instead, probably because of technology limitations at the time. TAME was used to create two small geo-annotated corpora called TR-CONLL (containing 946 documents) and TR-MUC4 (containing 100 documents), the first one using two annotators and manual adjudication of annotations (disagreement reconciliation), and the second one using only one annotator [5]. To the best of our knowledge, these corpora are not publically available.

In 2008, a subset of the ACE 2005 English corpus was annotated using SpatialML, a mark-up language for annotating generic (e.g., city) and proper (e.g., Washington) place names, as well as the paths between places [23]. The corpus consists of news and broadcast conversations of an international scope, containing 1014 articles with an average of 50 toponyms in each, and was annotated using the desktop Callisto annotation tool. Annotators manually looked up coordinates (using other resources) for each place name to enter into text fields in the tool. The resulting corpus is only available for a fee, and therefore is not conducive to open research.

The Edinburgh Geo-annotator is a more recent effort in creating a geo-annotation tool that allows users to resolve place names to toponyms in GeoNames [26]. For geo-annotation, users have to click on each place name in text one-by-one to view and assign a toponym to it on the corresponding map view (showing toponyms on the map for each place in the text one at a time). Further, the toponyms are not labeled on the map, requiring users to click on each of the many candidate toponyms to see the toponym name and feature type, another potentially time-consuming operation, and the ranking of toponyms (from most likely to least likely) is not clear. We are not aware of any corpus created using the Edinburgh Geo-annotator.

DeLozier et al. (2016) documented the most recent attempt at implementing a web-based geo-annotation interface, called GeoAnnotate (<https://github.com/utcompling/GeoAnnotate/>). Notably, it was used on a subset of the War of the Rebellion corpus (WOTR) [6,27]. Therefore, we refer to this tool as the WOTR GeoAnnotate to minimize confusion. While the interface allows for the modification of different types of named entities in the text (place, person, organization), it only allows annotators to add a location for each in the form of point or polygon geometries, and does not allow resolving place names to toponyms as possible in a gazetteer. This choice might have been an intentional one, given that historical (especially small landmark) place names might not exist in a standard gazetteer. Nevertheless, the tool requires human annotators to look up places using internet search engines (or other sources) and manually find coordinates for those even though many places named in the archives do exist in gazetteers.

Geo-annotation with only geometry (and not a toponym ID) limits the use of a corpus in any linked data-driven approach, however, the resulting corpus, which is publically available, can still be used in geoparsing evaluation using distance-based metrics [5,28]. Another remarkable feature of the WOTR GeoAnnotate tool is that it allows for “document level” geo-annotation, i.e., geo-annotating a document scope (geographic focus) as a whole with geometry, a feature that we have not seen in any other geo-annotation tool.

TAME, Edinburgh Geo-annotator and WOTR GeoAnnotate present users with a hardcoded list of place names (text spans/segments) already identified in the text, by either automatic NER or a human-annotator from another annotation project. Aside from erroneously annotated place names, existing place name corpora may carry the common and frequent error of considering entities to be references to places when they are actually metonymical (such as “Washington decides to ratify new sanctions”) or demonymical (such as “Chinese goods are flooding the markets”) uses of place names. These errors should be corrected for [6,29]. The ACE SpatialML corpus [23] and the Local Global Corpus (LGL) [2] also include these errors systematically, as evidenced in their annotation guidelines and documentation. However, the existing geo-annotation tools do not give human annotators the ability to correct mistakenly-annotated place names in text. This is especially problematic in tools that use the imperfect output of NER as the input to the geo-annotation task (instead of human-annotated



place names), such as the Edinburgh Geo-annotator. While a combination of different NER outputs may provide better initial NE pre-annotation [30], none of the tools reviewed in this section rely on more than one NER system. The best NER systems offer an approximate recall of 92% and precision of 88% on corpora of news stories [30], and we argue that human annotators should be able to correct NER generated pre-annotations prior to carrying out the toponym resolution step in order to produce a robust corpus. The correction of identified NE in text, however, requires the geo-annotation tool to be able to dynamically query the gazetteer with the newly identified NE, which is another missing feature from the existing tools.

None of the geo-annotation tools reviewed here offer a mechanism for managing input from multiple annotators or reconciling annotation disagreements, and none offer any capability for cross-validating different document parts (such as paragraphs) for annotation agreement. What is more, none of the publications documenting geo-annotation tools provide design recommendations to expedite the annotation task beyond the use of one (imperfect) NER system as an input (without methods to alleviate the incompleteness of input in an efficient manner), or ways to cope with special characteristics of place names in text or toponyms in gazetteers. For instance, a place name in text may refer to more than one toponym from a gazetteer, e.g., mountain ranges and rivers are often encoded in gazetteers such as GeoNames with multiple separate locations to provide a rough approximation of the linear feature. The existing tools do not provide a way of resolving place names to more than one toponym.

### 2.3. Iterative Design with the Designer as a User

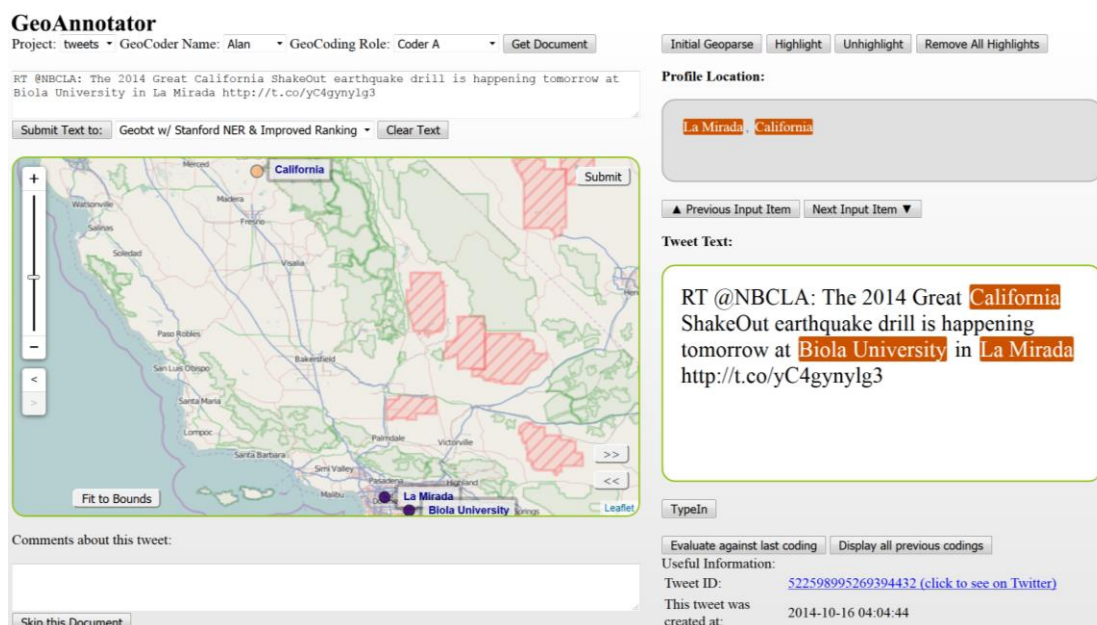
Previous studies have explored and experimented deconstructing the user-designer dichotomy. Some have studied the phenomenon from the point of view of the social roles taken by either group and the resulting power-relations, and how one group configures the behavior of the other one [31,32]. These studies argue that the boundary between user and designer is fluid and configured during the design process, and that users can have multiple identities [32]. In addition to being users, they can perform activities traditionally ascribed to designers by dynamically participating in the ongoing design process. Fleischmann (2006) argues for a user-centered design that deconstructs the traditional power relationship between designers and users through role hybridization by creating an environment where users or designers are able to shift from one role to another, effectively belonging to otherwise two distinct groups [33].

The approach we took is comparable to the ideas outlined by Park and Boland (2011), who, from a practical point of view, present a “dynamic model of designer-user interactions during a design project” [34]. A core idea that the authors document is that designers can take the role of users, and users can participate in the design process, with constant interactions within and between the users and designers in order to combine and reconcile design ideas from the two groups of users and designers. We used an early version of GeoAnnotator to manually resolve crowd-worker identified place names to toponyms as part of the GeoCorpora project [3]. That initial implementation and application of what could be considered an alpha version of GeoAnnotator was the starting point for the designer-developer process reported on here, which has led to a substantially improved version of GeoAnnotator. Section 3.2 provides details about many of the iterative improvements to GeoAnnotator implemented throughout this designer-user development process.

## 3. Methods: Iterative Design

Figure 1 shows GeoAnnotator’s UI in its early stages, where it was conceived only as a complementary “geocoding” interface in which text with pre-annotated named entities was fed into GeoAnnotator for identifying the corresponding geographic location (i.e., toponym). In other words, place names were pre-delineated in the text and users identified the corresponding toponym on the map for each name. The platform also allowed rudimentary highlighting and unhighlighting of pieces of text to correct input pre-annotations, and then assigning a toponym to each selected name.

However, as users continued to geo-annotate text, many new requirements were identified through user comments on submissions. In what follows, we describe the iterative design pattern, the identified requirements of the tool and how they were addressed in the design process.



**Figure 1.** GeoAnnotator’s user interface (UI) main window during the initial stages of development. The panel on the right shows the text that is already annotated for named entities and fed into the platform. Place names are highlighted in orange. The panel on the left shows the map interface where highlighted named entities are mapped using a pre-generated list of toponyms through pre-processing. If the default toponym is not the correct one, users can click on it, retrieve a list of “alternative toponyms” and select the correct one. There is often some context that makes it possible for a human annotator to identify the correct toponym, either within the tweet itself, from the profile information and/or from URLs that may be included in the tweet. The interface provides a link to the original tweet on Twitter so that the annotator can search for more context if necessary.

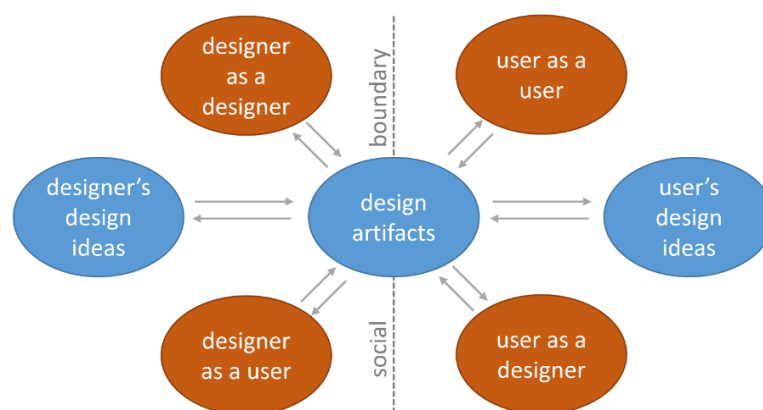
### 3.1. Iterative Design Pattern

We implemented the main UI and software architecture of GeoAnnotator according to insight gained from related literature and the needs expressed by our research group members: Undergraduate and graduate students and geography faculty members, who then used the platform to geo-annotate 2185 tweets from Twitter. Throughout the process of iterative design, we followed a “designer-as-user and user-as-designer” approach. Specifically, our work is grounded in the ideas presented by Park and Boland (2011), who describe designer-user interaction models as a combination of interaction of designers, interaction of users and mutual interaction between designers and users with constant communication to increase knowledge sharing, crossing the social boundaries and deconstructing the designer-user dichotomy to produce novel design artifacts and increase usefulness (Figure 2) [34].

The designers in the GeoAnnotator project were part of a larger team that was building a corpus to support the testing and training of GeoTxt [35]. Additional users who were contributing to the corpus-building effort also provided input on the tools and interacted with the designers and users in the conversational mechanism process outlined in Section 3.2.6.

During the first phase of the iterative design process (July 2015 to June 2016), we periodically modified or enriched the software and its UI according to the feedback provided by users in the form of explanatory comments submitted alongside annotations as well as through face-to-face meetings. In other words, the annotation task was integrated with a commenting/conversational mechanism, enabling user feedback on the missing annotation requirements for the specific document that raised the

need for the missing requirement. This enabled designers/developers to contextualize the comments of human annotators with the very specific annotation task at hand. These comments usually indicated one of two things: (a) GeoAnnotator's capabilities at the time did not support the annotation of that document or (b) that users were uncertain about how to geo-annotate some place names (such as West Africa), or how to deal with linguistic complexities. To address the latter point, guidelines were established to define what exactly constitutes a place name in text and what should be highlighted, and how, if necessary, special place names (such as ambiguous place names) should be geolocated to a toponym and what descriptor tags should be used to indicate such complexity/specialty. To support these guidelines and the assignment of special tags, GeoAnnotator was iteratively modified and expanded through and after the corpus building activity. During the second phase of iterative design and implementation, which lasted from June 2016 until August 2017, we expanded upon GeoAnnotator to further improve its usefulness.



**Figure 2.** Model of designer-user interactions in visual analytics tool design (after Figure 4 in Park and Boland, 2011, [34]).

The “designer-as-user and user-as-designer” iterative design pattern adopted here differs from both a standard engineering approach (in which users are not consulted until after a fully functioning system is implemented) or a “human-centered design” [36] process (in which iterative solicitation of target users’ work practices and needs are solicited at various stages). The key difference is that it integrates both input from target users who are not involved with the design and users who are also designers. This increases the likelihood that the developers understand the target user’s needs deeply while also supporting the potential to integrate design solutions that non-developers would be unlikely to think of (since they do not know what is possible). We elaborate on the expansion and modifications that we made as a model for related efforts to build visual tools to support annotation and as a means to put the decisions we made (and the end result) in context.

### 3.2. Feature Modification and Tool Expansion

In this section, we focus on modifications with important implications for manual geo-annotation as well as automatic geoparsing and spatial language studies, while avoiding mentioning specific, fine-grained changes made to the software to improve usability (such as moving or renaming or redesigning UI controls, components or icons).

#### 3.2.1. Global and Cross-Document Part Geo-Annotation

Not taking advantage of semi-automatic methods and redundancy in text can perhaps be one of the most important sources of inefficiency. Semi-automatic text highlighting/annotation techniques are not well-documented in the literature. The initial version of GeoAnnotator relied on manually pre-annotated text (in which place names were already identified). Later, we added the Stanford



Named Entity Recognition (NER) engine [37] to generate pre-annotations for named place entities on the fly. In the early stages of the project, we realized the limitation of relying on just one NER system, whose recall is approximately 92% at best, perhaps even lower for tweets [30]. This lack of complete recall results in some places not being highlighted in the pre-annotation. This is a considerable bottleneck in annotation, causing significant time delays spent reading every word, missed names, increased annotator disagreement and costly corpus building. Further, documents usually refer to the same names more than once, generating a redundancy that can be laborious for manual annotation unless harnessed for cross-control and validation. Therefore, we modified GeoAnnotator to present multiple parts of documents to the users at one time (two parts by default, such as the title and abstracts or two paragraphs) and applied the following modifications for maximum efficiency. Specifically, to minimize the likelihood of missing names in the text, we moved to an approach that integrates results from multiple NER engines as well as cross-document propagation and control.

The current version of GeoAnnotator leverages six NER engines: GATE ANNIE [38], CogComp NER [39], MIT IE (<https://github.com/mit-nlp/MITIE>), LingPipe (<http://alias-i.com/>) and Apache OpenNLP (<https://opennlp.apache.org/>) in addition to the Stanford NER engine. The result is a “Union NER”, in which the union of all place names detected by all NERs are incorporated in the pre-annotation process. Any character that is detected to be part of a place name by at least one NER is presented to the user as a candidate part of a place name in the initial annotation (as highlighted text). This process trades the avoidance of false negatives, which would be hard for users to notice and fix, for potential false positives, which annotators should be able to fix easily.

GeoAnnotator uses an innovative approach to enable annotating multiple parts of the document at the same time to take advantage of redundancy in documents and increase efficiency, speed and accuracy. To further prevent the possibility of missed places, for each place name that is detected by at least one NER in one of the document parts (e.g., title or abstract, user profile location or tweet text), all document parts are scanned for identical names (ignoring capitalization) using regular expressions, and those identical names are also added to the initial annotation as suggested (and highlighted) place names. This is especially useful for detecting place names in shorter document parts, such as academic article titles (e.g., to help recognize that “Washington” refers to a place, not a person, in “What Washington means by policy reform” [40]), or tweet profile locations that have a structure different from longer paragraphs. All NER engines in GeoAnnotator are trained (Stanford NER, CogComp NER, Open NLP, LingPipe and MIT IE) or adjusted (GATE ANNIE) for longer sentences and paragraphs, and they may miss place names in short text items that do not follow the capitalization pattern or grammatical structural of a regular sentence. Therefore, identical place names that are already detected in longer document parts (such as abstracts) are leveraged to automatically highlight places in the shorter parts (title). We refer readers to [30,35] for a recent comparison between the performances of these NER engines [30].

A similar global highlighting of place names is automatically performed for each place name that is manually highlighted by users, i.e., text in all document parts is scanned and similar place names are highlighted. When the human annotator chooses to unhighlight a (wrongly identified place) name, the system prompts them to choose between unhighlighting that individual name or unhighlighting all identical names in the text with one click.

We applied the same logic for cross-document part propagation (described above for place entity recognition) to geo-annotation. GeoAnnotator shows the assigned toponyms for each document part one at a time, i.e., the map renders all the places in the title or the abstract, and users can freely switch back and forth between different document parts. When a user changes the toponym for a place name on the map for any document part (e.g., title or abstract), the same correction is made for the other document part. This method of leveraging redundancy in document parts increases efficiency and gives users a chance to see the corrections they have made again in the second document part, meaning that users will get to review their corrections one more time before submission.

### 3.2.2. Advanced Toponym Search

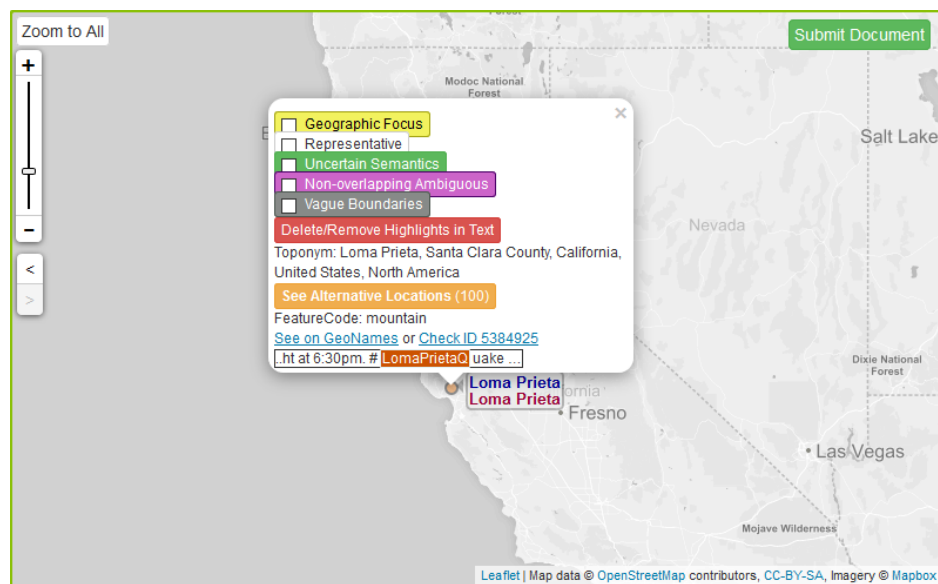
When a close variant of a place name as it appears in text (but not the full name, such as NYC for New York City) is not listed in a toponym's alternate names in the gazetteer (each toponym in GeoNames lists an array of nicknames, historical names and alternate names), the correct toponym might not be retrieved for users to select from. Incomplete names and disconnected qualifiers in conjunctively connected names such as "Black" in "Surrounded by Black and Caspian Seas" might suffer from the same issue. To address this issue, we initially added a "type in" feature (a simple text box) into GeoAnnotator, which users could use to type in a name for a highlighted place name in the text. However, we later realized that there were many instances of names shared between different feature types, such as "Mount Shasta", which is the name of both a mountain and a town. Therefore, we upgraded the "type in" feature with a more intuitive "advanced search" capability (Figure 3). Users can type in an alternative name and optionally select a feature class (type), country or continent to retrieve a list of 100 toponym candidates, meeting the filter constraints, that are ranked by their search score.

**Figure 3.** Advanced search for looking up places whose name in the text is not sufficient for the system to retrieve a list of relevant toponyms.

### 3.2.3. Special Tags

During the iterative design process, a periodic review of users' comments and examination of reported documents revealed several cases of user uncertainty due to (a) linguistic complexities of identifying place references, (b) spatial ambiguity or vagueness of some toponyms and (c) gazetteer incompleteness. These cases were reviewed and special tags were added to GeoAnnotator for users to mark such cases for potential special treatment in subsequent use of the corpora generated (Figure 4).

We have previously reported on these special tags in our earlier article that focuses on the linguistic analysis of the corpus created using GeoAnnotator [3]. Table 1 provides a summary of special tag descriptions and examples.



**Figure 4.** Adding special tags to a toponym by invoking the toponym popup.

The application of special tags for annotation reduces user confusion when they encounter boundary cases (e.g., “London is rethinking tariffs”) by providing them with the tools to mark such cases. Toponyms that are marked with a special tag (e.g., uncertain semantics) in the resulting corpus can be used for further research or excluded from algorithm evaluation.

Not all place names that appear in text have an entry in the gazetteer. These names include (a) places that are in general not listed in the gazetteer of choice, for which a different geographic knowledge base is needed (such as street addresses), (b) places whose boundaries are so vaguely defined that adding them to GeoNames is not justified (such as “Pacific Coast” in “the Ecuador Pacific Coast”), or (c) toponyms that have not been added to the gazetteer yet (but should be added).

In our corpus building activity, we devised one method of addressing this issue, but later on realized that it was extremely time consuming and that the end-result was not very useful in the subsequent geoparsing training and evaluation. We kept the functionality in the system (while enabling a viable approach explained below), and therefore, we detail our initial approach and its issues.

We initially added a “representative” tag, so that users could mark a toponym that is either spatially approximate to the location of the named place or contains it. For this to be consistent, users could designate the lowest level geographic unit that can be specified as containing the missing feature (e.g., the smallest administrative division, such as the containing county as opposed to the containing state). Our initial reasoning was that representative toponyms can be used in evaluating geoparsers based on distance error, i.e., the geographic distance between the location of the representative toponym and the predicted toponym. However, it proved increasingly difficult and time-consuming at times to locate appropriate approximate toponyms when annotating (e.g., what is approximate for a street? What is approximate for a vague region? What toponym contains a region such as Southern California?). Therefore, we added the “not in locations database” special tag, which means that no toponym is assigned to the identified place name. Upon applying this tag, however, the system forces users to first perform an advanced search (and potentially use alternative spellings along with optional feature codes such as “mountain” or “populated place”) to ensure that an appropriate toponym for the place name actually does not exist in the gazetteer.

**Table 1.** Special tags in GeoAnnotator to demarcate special cases of toponyms. For each special case, an example is provided with the place name underscored.

Tag Name	Application	Example	Notes
<b>Uncertain semantics</b>	When it is unclear if a name is in fact a place name, attribute, reference to an organization, or a boundary case with mixed word sense. The uncertain semantics tag enables corpus end users to include, exclude or isolate such cases for different research studies.	The rising violence by <u>Rikers Island</u> correction officers . . .	Rikers Island can be interpreted to refer to the island or the Rikers Island Correctional Center Facility (both of which are “places”) or to the prison as an organization, as well as a noun adjunct modifying “correction officers”.
<b>Vague boundaries</b>	When the place name refers to an area or region whose boundaries are not clearly agreed upon.	Temperatures in <u>Hudson Valley</u> . . .	Sources indicate that there are differences in opinion on the exact bounds of Hudson Valley.
<b>Not in gazetteer</b>	When the place name in text does not exist in the gazetteer yet, or is so vaguely defined that addition to gazetteer is not justified.	Headed to the <u>West Coast</u> .	Explained in more detail in the following paragraph.
<b>Overlapping ambiguous (always including human annotator assigned surrogates list, enforced by the system)</b>	When human annotators cannot confidently determine which one of multiple candidate toponyms that overlap in space is being referred to. GeoAnnotator allows users to assign multiple toponyms (i.e., surrogate toponyms) to the place name, and apply the “overlapping ambiguous” tag to indicate that these toponyms can interchangeably be used as the resolved toponym for that mention of Lagos (or any other similar situation).	A man just died of Ebola in <u>Lagos</u> .	GeoNames lists three toponyms for “Lagos” in Nigeria: Lagos State (administrative region), Lagos (section of populated place—the city that is within Lagos State) and Lagos Island (within Lagos City, which is within Lagos State). These entities have overlapping geospatial positions and all can be correct assignments.
<b>Non-overlapping ambiguous (with surrogates list)</b>	When human annotators cannot determine which one of multiple candidate toponyms that do not overlap in space is being referred to. Users can assign a surrogate list of potential candidate toponyms to a place name and apply the “non-overlapping ambiguous” tag to indicate that these toponyms can interchangeably be used.	<u>Washington’s</u> changing demographics.	Washington may refer to “Washington D.C.” or “Washington State”, for example. These toponyms do not overlap and it is unclear which one the text author originally meant to refer to.
<b>Non-overlapping ambiguous (without surrogates list)</b>	When human annotators cannot determine which one of numerous candidate toponyms (that do not overlap in space) is being referred to, and there are too many potential candidates to assign as surrogates. Users can select a potential toponym and apply the “non-overlapping ambiguous” tag without providing a surrogates list (making such cases distinguishable to corpus users, who may exclude or use the cases for special studies).	<u>Springfield</u> feels like spring!	Without additional context, Springfield may be referring to numerous toponyms in different geographic regions.

### 3.2.4. Toponym Differentiation on the Map

To enable users to examine a toponym on the map view at first glance, a textual label is attached to each symbol (which is surprisingly missing from other geo-annotation tools, requiring a click interaction from users). Place name highlights in text are visually linked to toponyms on the map using a brushing technique: Upon hovering the cursor over a text highlight, all other occurrences of the same name as well as the corresponding toponyms on the map will be highlighted using a different color and vice versa.

Throughout the iterative design process, users explained that they had to zoom in or out to make sure a particular toponym was located inside a state or country. To alleviate this problem, the spatial hierarchy of each toponym is displayed inside the toponym's popup, in the form of toponym, state, country or continent (Figure 5).

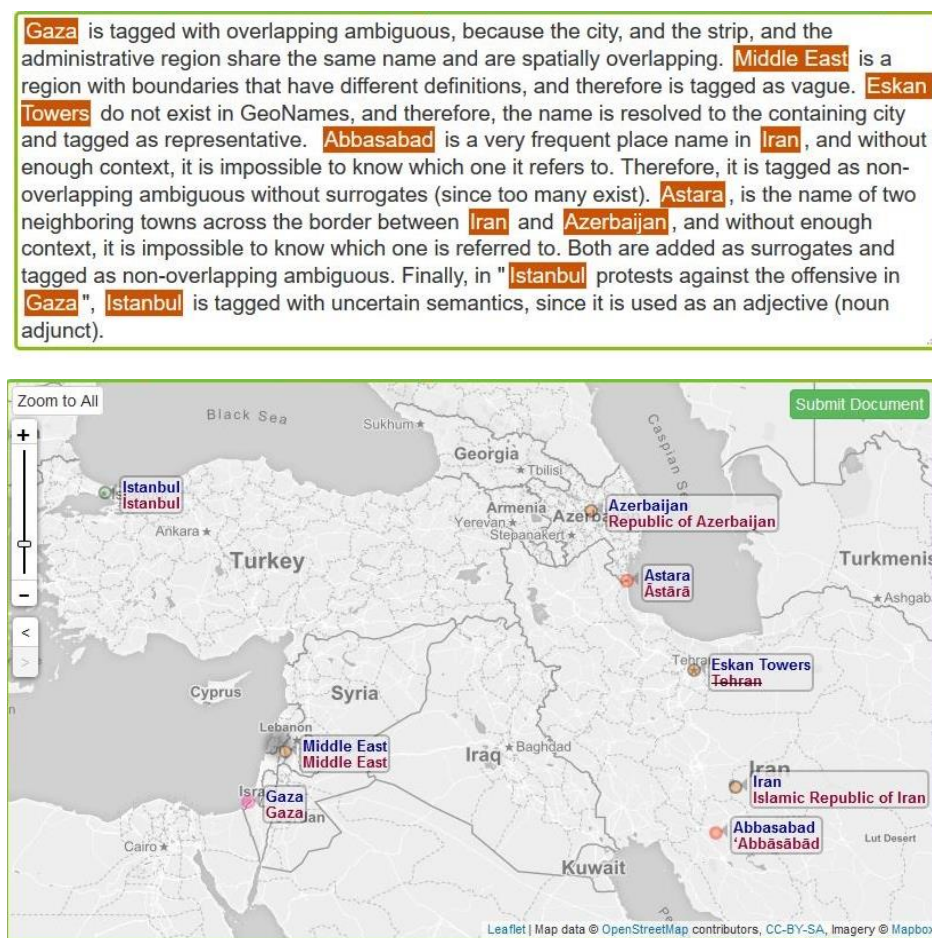


Figure 5. Disjoint map symbols for special tags.

More importantly, the designers who participated in the process as users noticed that many annotation disagreements arose from selecting spatially approximate candidates from different geographical levels or from different feature types that shared the same name. For instance, annotators may select “Tehran” as the province instead of “Tehran” as the city, even when contextual clues make it clear that the city is referred to. To address this issue, the GeoNames feature code is also included in the toponym popup (e.g., “capital of political entity” for Tehran, the city; or “first-order administrative division” for Tehran, the province). Using the feature code, annotators can distinguish between the two toponyms quickly.



### 3.2.5. Toponym Map Symbolization

After adding multiple special tags to toponyms, we modified map symbols so that users could quickly identify what tags are assigned to toponyms. Each toponym can have multiple tags at a time, therefore, we use several distinct symbol characteristics for each tag.

Figure 5 shows how the variations in symbolization and labelling reflect the presence of each tag. Toponym location is signified by a transparent circle (to allow overlapping symbols to be seen). Attributes of the circle and its surround signify the following: (a) Circle fill with orange hue—no tags, green hue—tag of “uncertain semantics”; (b) circle border with light gray hue—no tags (redundant with above), red hue—non-overlapping ambiguous with no assigned surrogates, red hue and double-ringed—non-overlapping ambiguous toponyms with surrogates, purple hue and double-ringed—overlapping ambiguous with surrogates (overlapping ambiguous toponyms always have surrogates); (c) circle background/shadow with fading borders—vague toponyms; and (d) toponym strikethrough—representative toponyms. It is worth noting that overlapping and non-overlapping ambiguous tags cannot coexist.

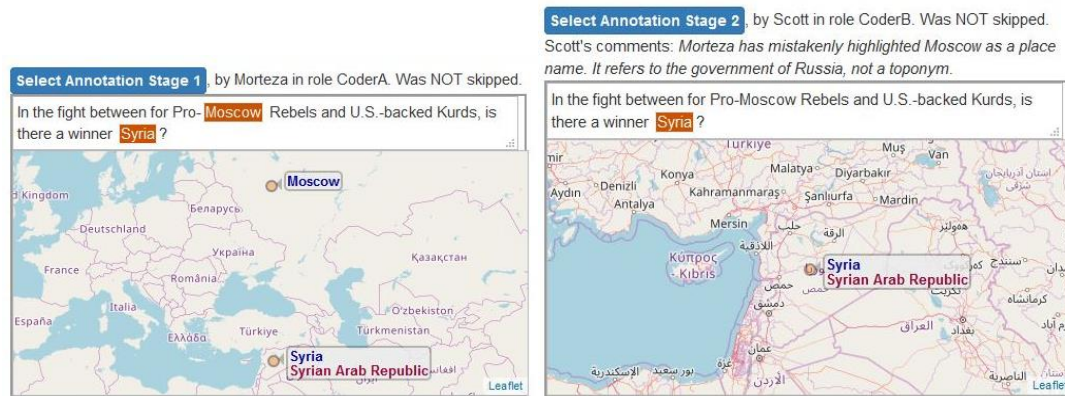
### 3.2.6. Conversational Mechanism and Annotation Agreement Criteria

To support reliability as well as iterative design while performing the actual task of annotation, each document is assigned to two annotators (called coder A and coder B below) for coordinated annotation with iterative, remote discussion until agreement is reached. The evaluator module of GeoAnnotator compares coder A’s and coder B’s annotations to determine agreement. For two annotations to be in agreement, the following properties have to match between the two: The specific characters highlighted in the text as a named place, the assigned toponym (determined by the GeoNameID) and any tags assigned to the entity (e.g., related to surrogates, ambiguity, vagueness, etc.). Upon submission by each coder, the system checks for agreement against the last submission by the other coder for the same document. If the annotations are in agreement, the evaluator changes the corpus status field of that document to “finished”. If the new submission does not agree with the previous one by the other coder, the system immediately alerts the coder of the mismatch, for potential reconsideration. If the coder insists on their annotation and there is still disagreement between the two coders’ annotations, the evaluator will assign the corpus a status of “annotation required”, and the corpus exposor module will send that document back to both coders one by one in order of initial annotation, giving them a chance to reconsider their annotation and to leave some explanatory comments on the annotation describing their reasoning, until their last annotations are in agreement with each other, as explained below.

When a document with annotation disagreement is presented to coder A or B again, the user can see their own as well as the other coder’s annotations in the form of highlights in the text and toponyms on maps, in addition to the comments coder A or B may have left on each submission (Figure 6). This enables a remote “conversation” between coder A and coder B about the reasoning for their choices. If both coders insist on their own annotations without agreement after three tries each, or if both coders report the document (using the report functionality, because they are either unsure how to annotate, or the software capability is not sufficient), the evaluator marks those documents with “uncertain” or “paused” corpus statuses, respectively. These documents are saved for future investigation and design recommendations.

The conversational mechanism outlined has three important outcomes. First, as indicated above, it enhances the reliability of end results by requiring agreement between two coders to add an entry to the corpus and by prompting coders to provide reasons for their annotations. Second, it helps establish consistent guidelines in geo-annotating documents. This outcome, an iterative development of guidelines, is documented in [3] for tweet geo-annotation. In that specific proof-of-concept application of GeoAnnotator, the iterative process drew attention not only to the distinction between overlapping and non-overlapping ambiguity among places that share a name but also to the challenge of polysemy (e.g., the use of a place name to mean the government of the place).

Third, as discussed, the conversational mechanism helps users provide feedback toward iteratively improving the functionalities and usefulness of GeoAnnotator. It is worth noting again that designers participated in the annotation task, closely involved in these conversations via the tool, and therefore, modified the system throughout and after the annotation cycle. The ability of the system to send back documents to a queue or pause (through “skipping while commenting” documents) made it possible to make design changes to meet specific requirements without halting the annotation task.



**Figure 6.** Annotation history window. Coders (users) can see previous geo-annotations of the same document (along with submitted explanatory comments) by themselves and the other coder of that document. The “select annotation” button populates the main window with annotations from any stage.

### 3.3. Case Study and Generated Corpus

The initial motivation for creating the GeoAnnotator system presented in this article was the need to construct a geo-annotated corpus of tweets in order to train and test the GeoTxt geoparsing system [35]. In a prior paper we detailed the construction of that corpus and its spatial linguistic analysis, called GeoCorpora, consisting of 6711 tweets, 2185 of which contain place names that were geo-annotated [3]. Here, we summarize the results of that corpus building effort, with a distinct focus on the key components of the process that framed the iterative development of the GeoAnnotator software detailed here.

During the informative phase of the iterative design of GeoAnnotator, we built a corpus of 2185 geo-annotated tweets (each 140 characters maximum in length) called GeoCorpora. The 2185 tweets that were ultimately included in GeoCorpora were pre-screened through an initial crowdsourcing process on Amazon Mechanical Turk (mturk.com) to ensure that they contained at least one place name. Users (undergraduate, graduate students and faculty members in geography) geo-annotated places in the tweet profile location as well as the tweet text (body). This process ran from July 2015 to June 2016, and the corpus took more than 250 man-hours of work to complete. Tweets posed an especially challenging task for annotation because of reasons such as abbreviations, misspellings and inadequate context. A total of 225 emails were exchanged (in addition to many meetings), and a total of 2965 comments were submitted alongside annotations. These conversations were used to iteratively improve the system and establish geo-annotation guidelines, during and after the corpus building activity. Also, during this time, we used the in-progress and incomplete corpus to optimize the internal geoparser of GeoAnnotator that generates the pre-geo-annotations. Users expressed that the geoparser improvement appreciably improved their performance (from approximately 30 documents per hour up to 50) and reduced the number of disagreements between coders due to toponym selection mistakes. Geoparsing optimization was primarily accomplished by configuring toponym ranking parameters based on population, alternative name similarities and assigning different weights to different GeoNames feature codes (such as capitals, physical features and populated places).

Each document was assigned to two annotators (following the logic explained in Section 3.2.6), who iteratively submitted their annotations until agreement was reached between the two. A total

of 5555 annotation submissions were made for the 2185 documents. One-hundred and sixty-three submissions were “report” submissions, meaning that GeoAnnotator’s capabilities at the time were not sufficient for annotation or that the guidelines needed reconsideration. As mentioned above, 2965 of the 5555 submissions included comments on either (a) reasons for disagreement or (b) suggested changes to the tool or guidelines. Table 2 lists the number of submissions and comments for each of the 6 annotators.

**Table 2.** Number of total annotation submissions and number of annotation submissions including comments.

Annotator	Number of Submissions	Including Comments
Faculty member	2515	1442
Graduate student	1698	830
Faculty member	883	442
Undergraduate student	156	72
Undergraduate student	303	179

Table 3 summarizes the number of submissions required for each tweet before geo-annotation agreement was reached (including assignment of tags, for places both in the profile location and text).

**Table 3.** Shows the count (X) for the number of submissions required for each tweet before agreement was reached.

X	2	3	4	5	6
Count	1335	576	219	48	7

Table 4 shows the frequency of place names in tweet text and tweet profile location. Almost 70% of tweets contained only 1 location in the text.

**Table 4.** Frequency of tweet documents with X number of place names in the profile location and tweet text.

X	1	2	3	4	5	6	7	Mean	Median	Mode
Profile location	676	573	66	11	1	0	0	1.56	1	1
Tweet text	1491	472	119	29	9	1	1	1.40	1	1

Table 5 summarizes the frequency of special tags assigned to toponyms in the generated corpus, both for profile locations and the tweet text. Not surprisingly, tweet text includes a much higher proportion of place names tagged with uncertain semantics compared to tweet profile locations. It is interesting, however, that profile locations needed fewer representative toponyms (i.e., more toponyms named in the profile locations are listed in GeoNames).

**Table 5.** Frequency of special tags and total number of place names in the corpus.

Place/Toponym Type	Tweet Profile Location	Tweet Text
Total place count	2069	2966
Uncertain semantics	9	178
Non-overlapping ambiguous	34	20
Overlapping ambiguous	60	91
Vague (excluding representative)	14	52
Representative-non-vague	28	150
Representative-vague	38	75

While tweets from the U.S. are more frequent than those from any other country (an interactive map detailing this can be found via this link: <https://resources.geovista.psu.edu/GeoCorpora/viz/>), it is clear from the map that the dataset used in the case study that drove GeoAnnotator development is global in coverage. Thus, the issues encountered and their impact on tool design within the iterative development process should be representative of those that might be encountered in future applications of GeoAnnotator, regardless of the geographical focus. That said, most tweets in the case study were in English, thus, it is likely that some modification to GeoAnnotator will be needed if the platform is to support other languages.

## 4. Results

### 4.1. System Overview

GeoAnnotator supports the quick and precise geo-annotation of place name entities (NE) in multipart textual documents such as scientific articles, with document parts including titles, abstracts and body text, or tweets, with parts consisting of the user profile location as well as the tweet's text. We designed GeoAnnotator to present multiple document parts at the same time to users, so that they can perform annotation utilizing the linguistic and spatial contextual clues that might be present in other document parts. This has the potential to help users determine whether a phrase is in fact a place reference, and if so, which location it might refer to. Also, the multi-document part representation enables the system to leverage redundancy in text to reduce the user workload (as explained in Section 3.2.1).

The GeoAnnotator user interface (UI) main window has two panels. The left panel includes separate text containers for each document part (configured for two-part documents in the figure), with one being “active” at a time (Figure 7). While users can see all document parts on screen, the interface activates each document part one at a time to enforce the precise annotation of that document part and avoid annotator confusion. Users can switch between different document parts while performing the annotation.

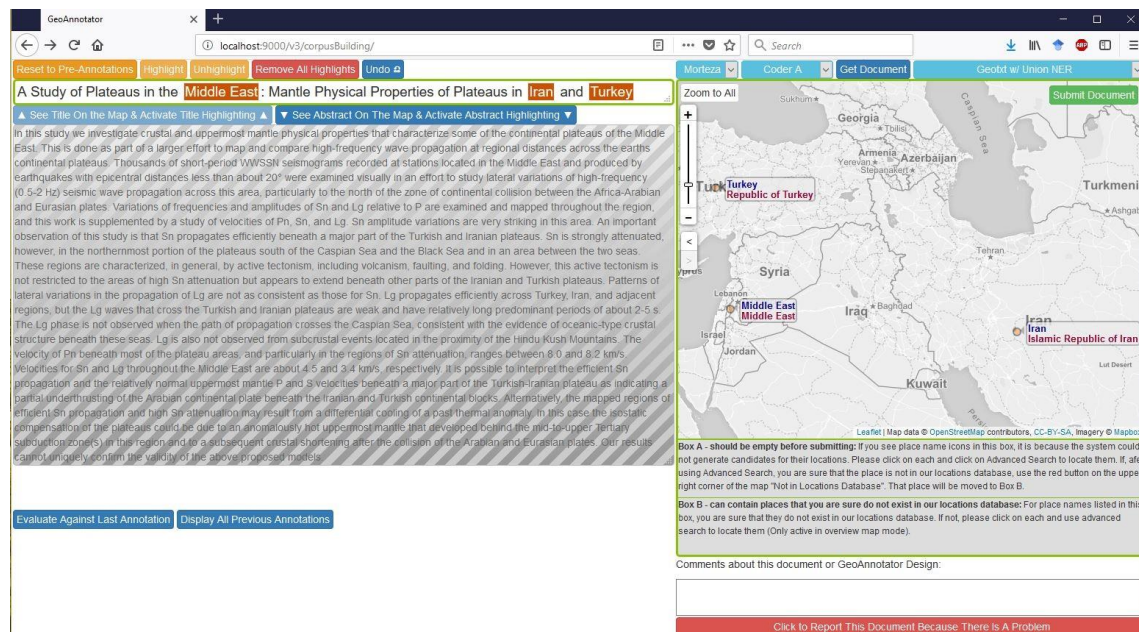
The GeoAnnotator user starts the annotation process by entering their login name and annotation role (explained in Section 3.2.6), and clicking on the “get document” button. The UI then loads a document from the database, along with the automatically generated pre-geo-annotation. This expedites the manual geo-annotation process, enabling the creation of large corpora at lower cost.

When the document loads, GeoAnnotator activates the first document part (in the case of Figure 7, the article title), and displays the pre-geo-annotation in the form of place names highlighted in the text, as well as a linked view of the proposed matching toponyms pinpointed on the interactive map panel on the right. The user is then able to confirm or correct the automatically generated geo-annotation as needed. To modify a place name highlight in the text, the user can select a piece of text (by holding their mouse's left button, dragging on the text and releasing, identical to text selection using the mouse pointer in other applications), then click on the “highlight” or “unhighlight” buttons to respectively add or remove that piece of text as part of a place name. The system automatically updates the map by attempting to resolve the newly highlighted place name in the text to a specific toponym (using choices generated by a query to a global index, currently consisting of an instance of GeoNames indexed using Apache Solr (<http://lucene.apache.org/solr/>)), or removing the toponym from the map if the “unhighlight” operation was triggered by the user.

Once the correction of the geo-annotation of the first document part is complete, the user moves on to the second document part and repeats the same process. Any correction to the automatically generated name highlights in the text, or to matches between those entities and toponyms on the map, propagates to matching features (if any) in the second document part. This both speeds up the annotation process and gives users another chance to double-check the highlights and the corresponding visually-linked mapped toponyms. This configuration was designed to improve annotation quality by providing a layer of control without creating any additional work for users.



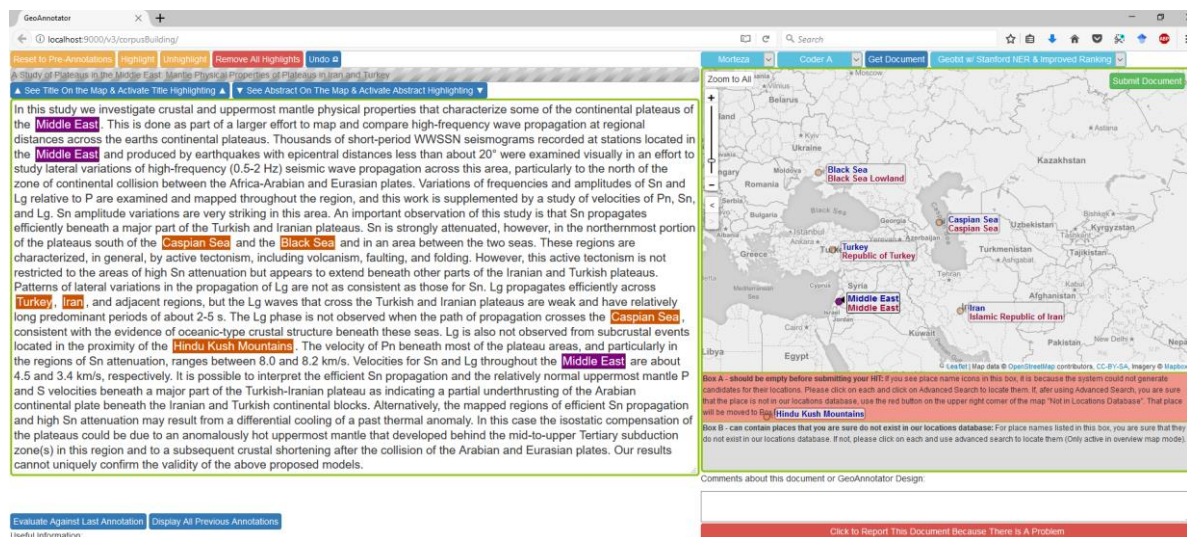
The annotator can switch freely back and forth between different document parts, as well as between correcting the place name highlights in text and correcting their toponym on the map. Place name highlights in text are visually linked to toponyms on the map using a linked brushing technique [41]: Upon hovering the cursor over the text highlight, all other similar names as well as the corresponding toponyms on the map are highlighted using a common highlight color, and vice versa. This visual link is essential in the annotation of longer pieces of text to speed up users' identification of toponyms that place highlights are resolved to (Figure 8).



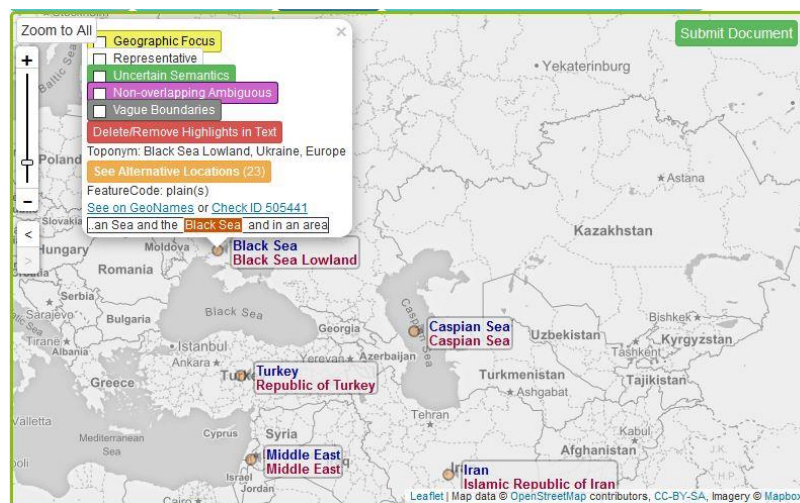
**Figure 7.** GeoAnnotator's UI main window layout, with the first document part (title) being active with place names highlighted in text (**left panel**) and mapped to the toponyms (**right panel**). The user can modify highlights or click on each toponym on the map to assign a tag or see alternative candidates. On the map, place names in text are labeled in blue, while the toponym official name from the gazetteer (sometimes identical, but often longer) is labeled in red.

In the map view, initially the toponyms with highest relevance (those that are computationally ranked as the most likely match) are assigned to each place name. The computational ranking, of course, is not always correct, which is the reason why a visual analytics interface is needed. For instance, in Figure 8, “Hindu Kush” is initially mapped to “Shewa Hindu Kush”, which is not the correct toponym referenced in the text. If the user decides that a place name is not resolved to the correct toponym, they can click on that toponym, then on the “see alternative locations” button in the popup that appears (Figure 9). The system then presents up to 100 alternatives toponyms for that place name in batches of 10, through which users can browse using buttons on the lower right side of the map view (Figure 10). Toponym candidates are presented in the order of descending ranks, therefore, the correct candidate is usually found in the first or second batch. Users can either select a candidate or use the advanced search functionality to retrieve more candidates based on a different name (or variation on the name) or more specific query parameters. After selecting a candidate, the map panel renders the overall view with the corrected toponym (Figure 11).

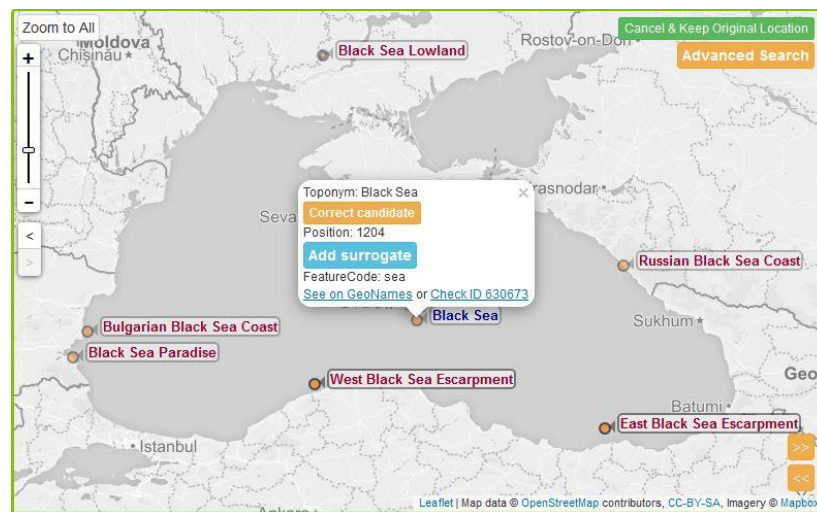




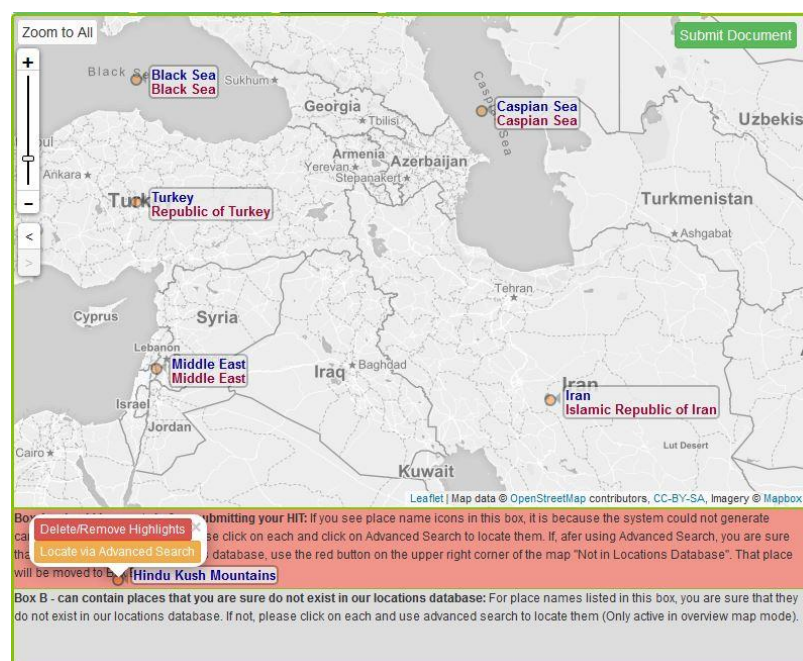
**Figure 8.** GeoAnnotator’s UI main window, with the second document part (abstract) being active for geo-annotation. Upon hovering the mouse pointer over the “Middle East” toponym on the map, all instances of “Middle East” in text will be highlighted in purple (and vice versa). “Black Sea” is not geolocated correctly, and no toponym is found for “Hindu Kush Mountains” by the system. The user can manually correct both of these issues.



**Figure 9.** Clicking on each toponym invokes a popup displaying a list of special tags, the spatial hierarchy of the toponym, its feature code (type) and link to entries on GeoNames for more information on each toponym. Users can use the “see alternative locations” button to see alternative toponym candidates for a place name.

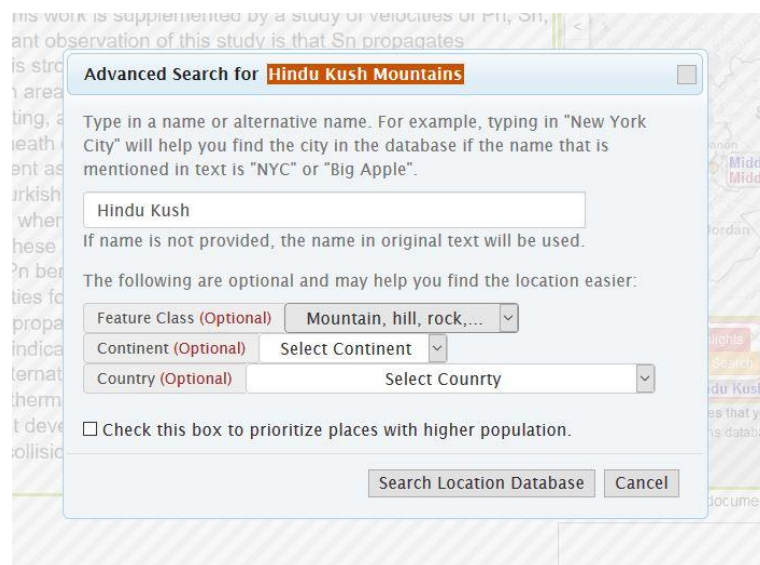


**Figure 10.** List of alternative toponyms for “Black Sea” displayed. The user can select one or more candidate as the ground truth toponym for a place name. If the correct toponym is not found in the list of 100 toponyms represented (in batches of 10), the user can use advanced search, accessible on the upper right corner of the map.



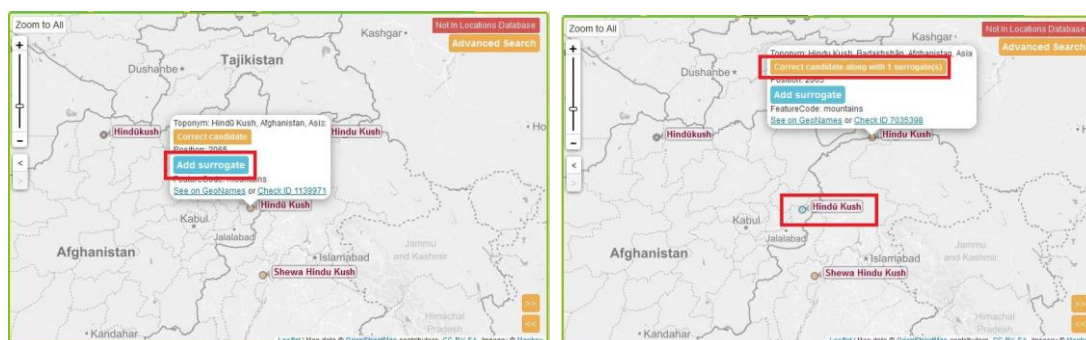
**Figure 11.** Overview of toponyms in text after correction for the Black Sea toponym. In the box highlighted in red below the map, it can be seen that “Hindu Kush Mountains” was not resolved to any toponym by the system. Box A’s color turned to light red to attract the user’s attention to this place name without an assigned toponym. The system does not allow document submission without the user performing an advanced search on such place names. This ensures that different spellings and variations of the place name are queried before deciding that it does not exist in the gazetteer.

When a correct place name is not retrieved through the steps above, as is the case for the Hindu Kush Mountains, in Figures 8 and 11, users can use the “advanced search” functionality (Figure 12) by providing name variants, features types or continents for a place name (in this example, the feature type including “mountain” is used to constrain the search).



**Figure 12.** Advanced toponym search. To retrieve more relevant toponyms for a place name, users can specify feature class, continent or country as well as by typing in a variant of the place name (such as New York City for NYC).

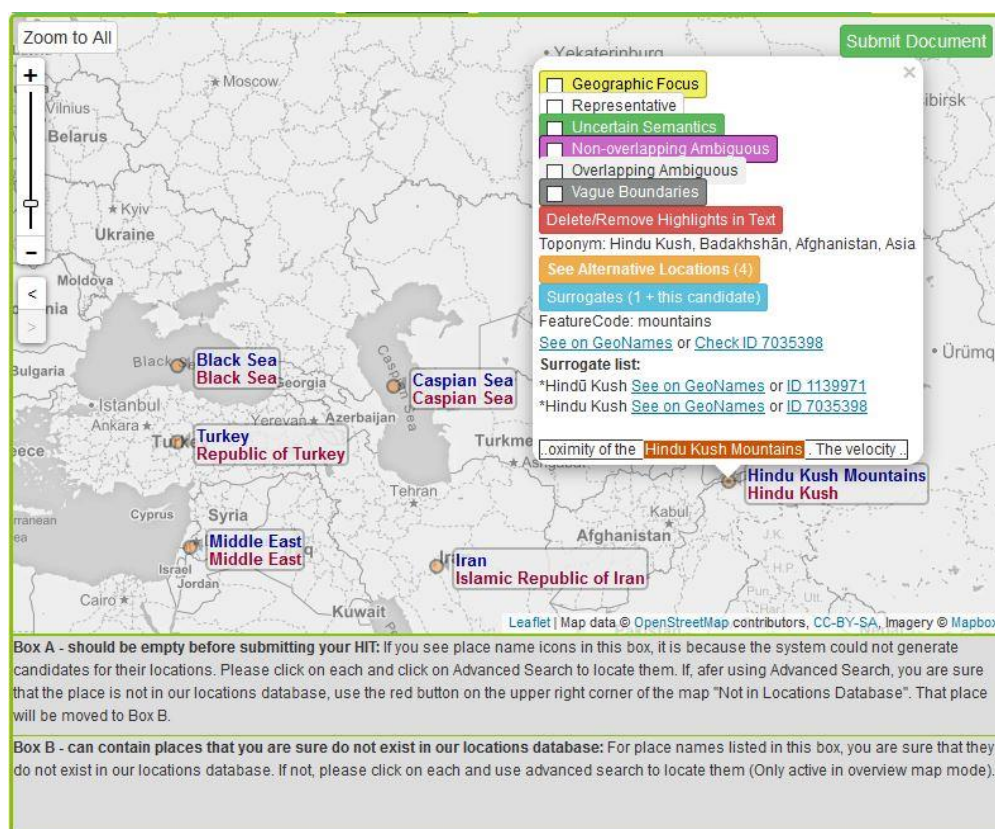
Hindu Kush is a mountain range that is near the Afghanistan-Pakistan border, and happens to have two toponyms in GeoNames, with locations at two different ends of the mountain range. GeoAnnotator allows users to assign more than one toponym to a single place name. For this, users can select one of the correct toponyms, click on “add surrogate” and click on the other one, and click on “correct candidate along with surrogate” (Figure 13). Surrogates are toponyms that, based on the information available, are equally appropriate matches to the place name. They are used, as in this case, to depict multiple specific locations used to specify an area or linear feature that has no single agreed upon point location, as well as to depict alternative possible matches for ambiguous cases (as discussed in more detail below).



**Figure 13.** Adding surrogates to a place name. Users can assign more than one toponym to a single place name. The “not in locations database” button allows users to specify that a place name is not in the gazetteer.

Figure 14 shows the resulting geo-annotation after adding two toponyms for the Hindu Kush Mountains. The symbol for that toponym changes to a circle with a ring to indicate nested surrogates for that place name. Users can click on the popup to see the list of surrogates or modify them.





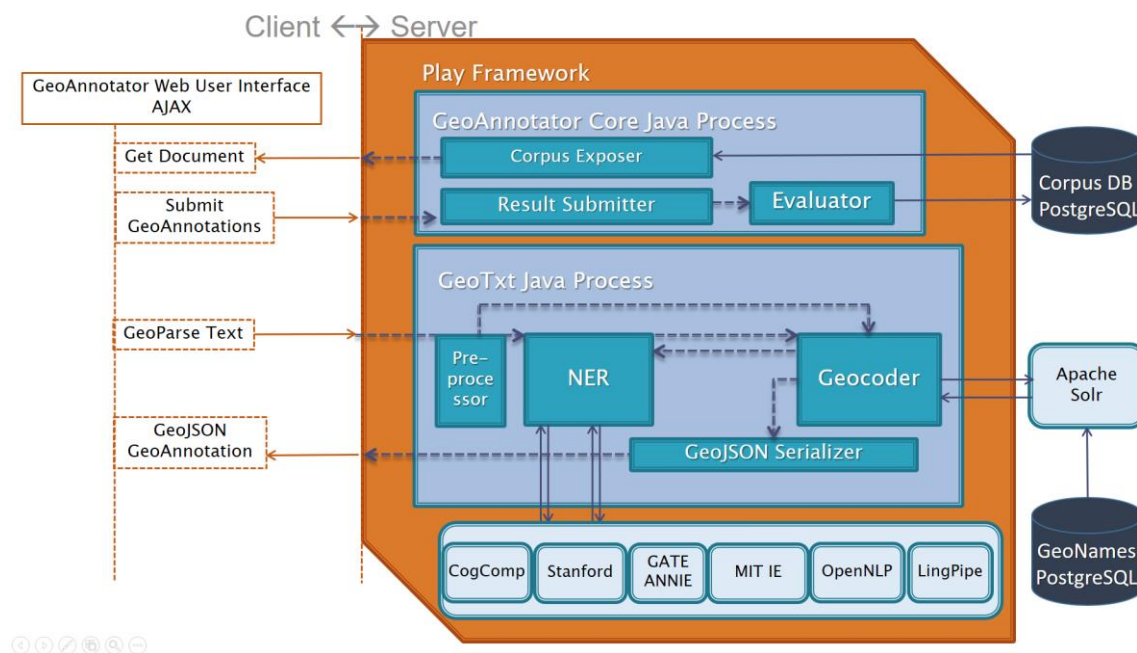
**Figure 14.** Geo-annotation after correction for the Black Sea and the Hindu Kush Mountains (which is now removed from box A under the map).

Users can submit documents as either completed or problematic. In the latter case, comments can be added to explain issues. These comments, along with the submitted geo-annotation, constitute input to the conversational mechanism of GeoAnnotator, which (as explained in detail in Section 3.2.6) enables those doing the annotation to collaborate on tricky cases and increase annotation reliability.

#### 4.2. Architecture

GeoAnnotator's software architecture is designed and implemented using interoperable interfaces. System components communicate using open application programming interfaces (APIs) so that each component can be replaced by a different technology if necessary. GeoAnnotator is implemented according to a two-tiered client-server model. The server is implemented in Java, and the client (UI) is implemented as highly interactive web pages. The UI is built using HTML5, CSS and JavaScript to be scalable and accessible on a web browser to many annotators. It utilizes Bootstrap (<http://getbootstrap.com/>) to be flexible on different display sizes, Leaflet (<https://leafletjs.com/>) for mapping and jQuery (<http://jquery.com/>), jQuery UI (<https://jqueryui.com/>) and Rangy (<https://github.com/timdown/rangy>) for operations and text highlighting to be fully functional across different browsers.

GeoAnnotator's server core consists of three modules and a PostgreSQL database (Figure 15). The corpus expositor module API accepts each annotators' login information and their selected role as input parameters and exposes documents retrieved from the corpus database (DB). The result submitter module's API takes the user submitted annotation in GeoJSON as its input parameter, parses the submitted object, and passes it to the Evaluator module, which compares the submitted annotation to previous annotations for the same document to determine the corpus status for that document, subsequently committing the annotation along with the determined corpus status to the corpus DB.



**Figure 15.** General architecture of GeoAnnotator. The Play framework is used to expose Java application programming interfaces (APIs) as HTTP web-service endpoints.

GeoAnnotator, as outlined above, is a semi-automatic annotation system. To make annotators' work easier and faster, the system first automatically geo-annotates a document by passing it through the API of the GeoTxt Geoparser [35] that is loosely-coupled with GeoAnnotator. GeoTxt is also utilized every time an annotator modifies the geo-annotation of a document. GeoTxt has two primary modules, one for NER and one for the geocoding of places recognized. The second can be used independently and this is taken advantage of by GeoAnnotator when a human annotator uses the "highlight" option. In these cases, highlighted text is assumed to be a place name and is sent directly to the GeoTxt geocoding module. GeoTxt is accessed via Ajax from the GeoAnnotator web UI rather than by the server, making it practical for other geoparsers to be used with no change to the GeoAnnotator server. Each place name query returns up to 100 candidate toponyms (as a GeoJSON feature collection), ranked by the likelihood of being the correct match.

Geo-annotations are stored as GeoJSON feature collections [42]. Each feature object in the feature collection array represents a place name in the text, whose position(s) in the document is (are) specified by an array holding character offsets of that name relative to the start of the text string. The character offset of place names is independent from stemming and tokenization algorithms implemented by different NER engines or geoparsers, and therefore, increases the usability of the corpus with various algorithms and tools with minimal token mapping. Each feature object also lists a number of properties, including the unique GeoNames ID of the toponym, the toponym name (such as United States) and the name in the text (such as US). The toponyms' GeoNames ID can be used to enrich features with additional information such as polygon boundaries (currently available with a premium subscription from GeoNames.org (<http://www.geonames.org/products/premium-data-polygons.html>)). Table 6 summarizes the features of GeoAnnotator and the other existing geo-annotation tools reviewed in Section 2.2.



**Table 6.** A comparison of the features of geo-annotation tools.

	GeoAnnotator	TAME	WOTR GeoAnnotate	Edinburgh Geo-annotator
Gazetteer	GeoNames	GNS/GNIS Toponym	N/A JavaScript Object	GeoNames
Output	GeoJSON	Resolution Markup Language	Notation (JSON)	Not reported
Map view	✓	×	✓	✓
Drawing custom geometries	×	×	✓	×
Assigning toponyms to place names	✓	✓	×	✓
Cross-document (pre)annotation	✓	×	×	×
Simultaneous named entity and toponym manipulation	✓	×	×	×
Special tags	✓	×	×	×
Non-place name entities support	×	×	✓	×
Multi-annotator Support	✓	×	×	×
Multi-toponym assignment to a single name	✓	×	×	×
Integrated NER for pre-annotation	✓	×	×	×
Advanced toponym search	✓	×	×	×
Named entity annotation manipulation	✓	×	✓	×
Document level (geographic focus) annotation	✓	×	×	×

## 5. Concluding Discussions: Insights from the Iterative Development Process

Here, we reflect on the user-designer/designer-user human-centered design process that was applied to the development of GeoAnnotator. We believe the iterative design of GeoAnnotator has lessons for user-centered software design. With inter-user communications capability built into the platform (in addition to the periodic face to face meetings and email exchanges), both users and designers were able to articulate their needs, comment on particular linguistic and geographic difficulties, deliberate together and suggest solutions for implementation. Moreover, the process of designing and implementing GeoAnnotator was not only focused on producing a geo-annotation software platform, but included a proof-of-concept effort to build a manually geo-annotated corpus along the way. The objective was to improve the annotation platform through insights gained by real use in building a corpus, without hindering corpus creation. Therefore, not only was it necessary for the software design process to be iterative, to ensure functional software for geo-annotation, but this iteration also had to be integrated with the actual use of the platform in the annotation of a corpus to define how geo-annotation for various and sometimes complex kinds of place references and toponyms should be performed.

Furthermore, geo-annotation requires either local knowledge of the place [5] or general geographical knowledge and the initiative and ability to further research the existence and location of places. In case of disagreement in geo-annotation between users, our strategy was to support communication and collective decisions among users to collaboratively establish guidelines for similar cases, rather than to just report disagreement rates between annotators (such as what was done in [5,23]) without understanding the reasons for disagreement (including even, perhaps, annotation software shortcomings). A corpus generated without this understanding and appropriate software may contain inherent biases due to assumptions made prior to considering real-world data. Therefore, we made a decision to use the conversational mechanism and annotation agreement criteria as a strategy to limit the potential for such biases. The communication among users through the platform revealed the challenges of toponym resolution, including common kinds of ambiguity that needed to be addressed at the software level. As detailed above, we also used this conversational mechanism to improve GeoAnnotator's capabilities over time. Annotators had the option to "report" a document that they found philosophically challenging to annotate or problematic because of technical shortcomings. The conversational mechanism allowed some of those reported issues to be resolved, but in cases where that was not possible, the reports provide evidence to help understand the complexity in place

references within languages and the challenges in matching place names to specific named entities in the world.

Consider the excerpt “everyone in Guam is hospitable”. Some geoparsers (or human annotators) may resolve Guam to the dependent political entity toponym for Guam (GeoNameID of 4043988), while others might choose the island toponym (GeoNameID of 4043542). If the intention of geoparsing is to create a spatial index or create geovisualizations, picking one candidate over the other one should not be penalized heavily since the locations are essentially the same (subject to sources of political versus topographical data). Such cases were identified through the conversational mechanism of GeoAnnotator, i.e., the process of users’ revisiting previous annotations of their own and by others. An “overlapping ambiguous” special tag was added, along with the capability for users to add “surrogates” to a toponym, which enabled both toponyms to be assigned for Guam (and similar situations). If a classic majority vote criteria or inter-annotator agreement threshold was adopted, many correct annotations (such as island of Guam versus the country of Guam) would have been considered as “disagreement” cases and discarded in subsequent use, without knowing the source for disagreement, leading to a waste of resources that would make building a corpus even more laborious and result in a corpus that did not include these important (and relatively common) cases.

The iterative refinement of GeoAnnotator resulted in a streamlined and comprehensive tool for geo-annotation that supports the special characteristics of place names and toponyms (such as overlapping ambiguous or vague toponyms). These special characteristics have implications for the design of geo-annotation platforms generally, the criteria used in determining annotation agreement, and beyond that, the design and evaluation of geoparsing algorithms. For instance, toponyms that could only be located using “advanced search” in GeoAnnotator (which are marked in the generated resulting corpus distinctly), are usually difficult cases for which a direct query of a gazetteer for toponym resolution has failed to yield relevant results. These are interesting cases to test a geoparser’s ability to harness context (to determine auxiliary query information such as feature type) in order to resolve to relevant toponyms.

GeoAnnotator is designed to support place name entity annotation in text, and also geo-annotation, i.e., manually segmenting and resolving place names to toponyms (with their respective geolocation). It can be configured to present text to annotators with no pre-annotation, so that annotators can manually identify place name entities and resolve to toponyms from scratch, or it can be configured to present an initial pre-annotation with named entities already identified by the union of multiple NER engines. Presenting place names to users with pre-annotation hastens the geo-annotation process but comes with a risk. It is likely, especially for larger pieces of text, that users might miss place references that are not already highlighted in text by the pre-annotation process, and therefore, these entities are not added to the map view and might be omitted due to visual absence. Therefore, we recommend using aggressive NER engines (those that prefer recall over precision) for the named entity recognition stage, or using ensemble methods that utilize multiple NER implementations to increase the likelihood of detecting place names in text. With aggressive pre-annotations, users are able to scan the text quickly and examine the context of each highlighted place name in the window around it for geo-annotation (without carefully reading the whole text, potentially), and remove erroneous highlights that are not in fact place names. Such corpora generated using pre-annotations can be suitable for use in spatial language research or optimizing toponym resolution algorithms.

Lastly, the concepts and techniques introduced here for automatic pre-annotation, global highlighting and cross-document part propagation/validation can be adopted in designing annotation platforms for other kinds of entities (e.g., persons or organizations).

## 6. Future Research

GeoAnnotator relies on its internal geoparser to predict geo-annotations for a document. The closer this prediction is to reality, the less modification is required by users.. Therefore, research on performant geoparsers is interdependent with research on improving semi-automatic annotation

platforms. Further, the quality of a geoparser's output annotation is domain-dependent. Depending on the kind of text (e.g., news story, social media posts or academic articles), the geographic coverage area of documents (e.g., North America, Middle East, Eastern Europe), and the geographic level of toponyms (e.g., country level, state level, physical features, landmarks or building names or addresses), the geoparser may underperform and therefore create additional work for users performing annotation. While one of the primary reasons for generating a geo-annotated corpus might be to develop better geoparsers, the incomplete and work-in-progress corpus can be used during the annotation process to improve GeoAnnotator's internal geoparser.

In subsequent research, we intend to use supervised learning algorithms to make the GeoAnnotator's geoparser "learn" from annotators' submitted annotations. This way, as more documents are annotated, the internal geoparser's performance and its resulting predicted annotations will improve in quality, which in turn will facilitate the creation of larger annotated corpora.

Unsupervised learning algorithms (clustering) may also be applied to documents' text, metadata or external knowledge bases to improve the initial annotation process. For instance, the system can potentially assign toponyms to clusters of topics using latent Dirichlet allocation [43], and once toponyms in the cluster are resolved by an annotator in a document, other toponym candidates within the cluster can be modified to suggest better initial geo-annotations in subsequent documents. Ju et al. (2016) provided an example of how topic models can be used to improve toponym resolution [24].

In future research, beyond improving pre-annotation, we are planning on using other custom special tags to model the spatial roles of toponyms in GIR research. For instance, we have added a "focus" tag (Figure 9) to the system. Users can assign it to indicate that a particular toponym is the focal place of a story, i.e., a place that a document is "about". A document might name many places in its textual content, but might primarily tell a story about only one or a few of them. Users will be able to use the "focus" tag to distinguish these places from others. Subsequent research will involve generating a corpus of manually annotated focal places, quantifying how (much) human annotators agree on the focal places of documents and examining whether algorithms can automatically determine the focal place(s) of the documents. Much like basic geoparsing research, the research field of "focal places" suffers from lack of publicly available benchmark datasets [44].

Custom special tags in GeoAnnotator can be used in spatial language research. For instance, GeoAnnotator can be used to create a corpus for spatial role labeling algorithms, the goal of which is to identify the spatial role of any place or named entity in text. Spatial role labeling is defined as "the task of identifying and classifying the spatial arguments of spatial expressions mentioned in a sentence" [45]. For example, in the sentence "fires were burning south of Carmel Valley", "fires" is referring to a trajectory object, the phrase headed by the token "Carmel Valley" refers to the role of a landmark and these are related by the spatial expression: "south of". Spatial role labeling is key not only in geographic information retrieval [46] but also in domains such as text-to-scene conversion, robot navigation or traffic management systems. The conversational mechanism of GeoAnnotator can help refine various spatial roles identified and defined for such tasks.

Future research should also explore ways in which geo-annotation can be crowdsourced to create large corpora in different domains to train supervised machine-learning algorithms. Volunteer citizen scientists can potentially be recruited if the geo-annotation process is (made) interesting enough to them that they either enjoy the process of annotation (for example, to score points as part of a larger competition, or gamification of the annotation process, as showcased by [14]), or they learn something of interest to them while performing the annotation, such as stories about the local geographies of different places in the world.

As our next immediate step, we will be addressing research questions necessary to test whether geo-annotation can be successfully crowdsourced on platforms such as Amazon Mechanical Turk (<https://www.mturk.com/>) or Figure Eight (<https://www.figure-eight.com/>). These platforms handle the recruitment of distributed paid human workers, which we call "crowd workers". Now that GeoAnnotator's capabilities are refined and proven to be functional, and also sources of geo-annotation

uncertainty are identified and annotation guidelines are established, we hypothesize that it will be possible to switch the annotation agreement evaluation criteria to a classic majority vote for use with large numbers of distributed annotators. To test this hypothesis, we will qualify the crowd workers by having them annotate a corpus that is already geo-annotated by expert annotators. We will compare the inter-annotator agreement of expert annotators and that of crowd annotators, and identify crowd workers whose performance is acceptable.

We will also investigate the development of rules that use annotations of the same document by different users to automatically assign special tags to toponyms: For example, to identify overlapping-ambiguous toponyms and to populate a list of surrogate toponyms for them automatically. To determine annotation agreement and the automatic creation of these tags, we will use spatial hierarchy, string similarity, feature type and the geographic coordinates of toponyms to automatically assign overlapping ambiguous tags and populate surrogate lists.

Overall, GeoAnnotator is part of a larger strategy to develop geoparsing systems that can quickly and accurately leverage the wealth of geographic data that remains hidden in textual documents. A key to advances in geoparsing is the development of more effective and efficient ways to build large corpora of documents in which place references are both recognized and resolved to real world locations. GeoAnnotator and the planned extensions outlined above offer a platform for progress in this domain.

**Author Contributions:** Conceptualization, methodology, and validation: Morteza Karimzadeh and Alan M. MacEachren; formal analysis, resources, data curation, resources, software, writing—original draft preparation, visualization, Morteza Karimzadeh; writing—review and editing, Alan M. MacEachren.

**Funding:** This work was partially funded by the U.S. Department of Homeland Security’s VACCINE Center under award number 2009-ST-061-CI0003.

**Acknowledgments:** We thank Scott Pezanowski for his set up of a Solr index on GeoNames data, and contributions to the GeoTxt geoparser leveraged in GeoAnnotator. We also thank Christina Stober for proofreading the final draft.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study, nor in the collection, analyses, or interpretation of data, nor in the writing of the manuscript, or in the decision to publish the results.

## References

1. Finlayson, M.A. Report on the 2015 NSF Workshop on Unified Annotation Tooling. 2016. Available online: <https://dspace.mit.edu/handle/1721.1/105270> (accessed on 17 March 2019).
2. Lieberman, M.D.; Samet, H.; Sankaranarayanan, J. Geotagging with local Lexicons to build indexes for textually-specified spatial data. In Proceedings of the 26th International Conference on Data Engineering (ICDE 2010), Long Beach, CA, USA, 1–6 March 2010; pp. 201–212.
3. Wallgrün, J.O.; Karimzadeh, M.; MacEachren, A.M.; Pezanowski, S. GeoCorpora: Building a corpus to test and train microblog geoparsers. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 1–29. [CrossRef]
4. Gritta, M.; Pilehvar, M.T.; Limsopatham, N.; Collier, N. What’s missing in geographical parsing? *Lang. Resour. Eval.* **2018**, *52*, 603–623. [CrossRef]
5. Leidner, J.L. *Toponym Resolution in Text*; University of Edinburgh: Edinburgh, UK, 2007.
6. DeLozier, G.; Wing, B.; Baldridge, J.; Nesbit, S. Creating a novel geolocation corpus from historical texts. In Proceedings of the 10th Linguistic Annotation Workshop Held in Conjunction with ACL 2016 (LAW-X 2016), Berlin, Germany, 11 August 2016.
7. Acheson, E.; De Sabbata, S.; Purves, R.S. A quantitative analysis of global gazetteers: Patterns of coverage for common feature types. *Comput. Environ. Urban Syst.* **2017**, *64*, 309–320. [CrossRef]
8. Bontcheva, K.; Cunningham, H.; Roberts, I.; Roberts, A.; Tablan, V.; Aswani, N.; Gorrell, G. GATE Teamware: A web-based, collaborative text annotation framework. *Lang. Resour. Eval.* **2013**, *47*, 1007–1029. [CrossRef]
9. Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L.; Weischedel, R. OntoNotes: The 90% solution. In Proceedings of the Human Language Technology Conference of the NAACL, Companion, New York, NY, USA, 4–9 June 2006; pp. 57–60.

10. Kim, J.D.; Ohta, T.; Tateisi, Y.; Tsujii, J. GENIA corpus—A semantically annotated corpus for bio-textmining. *Bioinformatics* **2003**, *19*, i180–i182. [[CrossRef](#)] [[PubMed](#)]
11. Kim, J.-D.; Ohta, T.; Tsujii, J. Corpus annotation for mining biomedical events from literature. *BMC Bioinform.* **2008**, *9*, 10. [[CrossRef](#)]
12. Sabou, M.; Bontcheva, K.; Derczynski, L.; Scharl, A. Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. In Proceedings of the Ninth International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 26–31 May 2014; pp. 859–866.
13. Cano, C.; Monaghan, T.; Blanco, A.; Wall, D.P.; Peshkin, L. Collaborative text-annotation resource for disease-centered relation extraction from biomedical text. *J. Biomed. Inf.* **2009**, *42*, 967–977. [[CrossRef](#)] [[PubMed](#)]
14. Dumitrache, A.; Aroyo, L.; Welty, C.; Sips, R.J.; Levas, A. “Dr. Detective”: Combining gamification techniques and crowdsourcing to create a gold standard in medical text. *CEUR Workshop Proc.* **2013**, *1030*, 16–31.
15. Apostolova, E.; Neilan, S.; An, G.; Tomuro, N.; Lytinen, S. Djangology: A Light-weight Web-based Tool for Distributed Collaborative Text Annotation. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10), Valletta, Malta, 17–23 May 2010; pp. 3499–3505.
16. Papazian, F.; Bossy, R.; Nédellec, C. AlvisAE: A collaborative Web text annotation editor for knowledge acquisition. In Proceedings of the Sixth Linguistic Annotation Workshop, Jeju, Korea, 12–13 July 2012; pp. 149–152.
17. Stenetorp, P.; Pyysalo, S.; Topić, G.; Ohta, T.; Ananiadou, S.; Tsujii, J. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics EACL ’12, Avignon, France, 23–27 April 2012; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 102–107.
18. Meurs, M.J.; Murphy, C.; Naderi, N.; Morgenstern, I.; Cantu, C.; Semarjit, S.; Butler, G.; Powlowski, J.; Tsang, A.; Witte, R. Towards evaluating the impact of semantic support for curating the fungus scientific literature. *CEUR Workshop Proc.* **2011**, *774*, 34–39.
19. Bontcheva, K.; Derczynski, L.; Roberts, I. Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*; Springer: Berlin, Germany, 2014; pp. 1–18.
20. Chinchor, N.; Sundheim, B. MUC-5 evaluation metrics. In Proceedings of the the 5th Conference on Message Understanding, Baltimore, Maryland, 25–27 August 1993; pp. 69–78.
21. Tjong Kim Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003—Volume 4 CONLL ’03, Edmonton, AB, Canada, 27 May 2003; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 142–147.
22. Ritter, A.; Clark, S.; Mausam, E.; Etzioni, O. Named entity recognition in tweets: An experimental study. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 1524–1534.
23. Mani, I.; Hitzeman, J.; Richer, J.; Harris, D.; Quimby, R.; Wellner, B. SpatialML: Annotation Scheme, Corpora, and Tools. In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco, 26 May—1 June 2008.
24. Ju, Y.; Adams, B.; Janowicz, K.; Hu, Y.; Yan, B.; McKenzie, G. Things and strings: Improving place name disambiguation from short texts by combining entity Co-occurrence with topic modeling. *Lecture Notes Comput. Sci.* **2016**, *10024 LNAI*, 353–367.
25. Leetaru, K.H. Fulltext Geocoding Versus Spatial Metadata for Large Text Archives: Towards a Geographically Enriched Wikipedia. *D-Lib Mag.* **2012**, *18*, 5. [[CrossRef](#)]
26. Alex, B.; Byrne, K.; Grover, C.; Tobin, R. A Web-based Geo-resolution Annotation and Evaluation Tool. In Proceedings of the LAW VIII—The 8th Linguistic Annotation Workshop, Dublin, Ireland, 23–24 August 2014; pp. 59–63.
27. The War of the Rebellion: Original Records of the Civil War. Available online: <http://ehistory.osu.edu/books/official-records> (accessed on 1 January 2018).
28. Karimzadeh, M. Performance Evaluation Measures for Toponym Resolution. In Proceedings of the 10th Workshop on Geographic Information Retrieval, GIR ’16, Burlingame, CA, USA, 31 October 2016; ACM: New York, NY, USA, 2016; pp. 8:1–8:2.



29. Gritta, M.; Pilehvar, M.T.; Limsopatham, N.; Collier, N. Vancouver Welcomes You! Minimalist Location Metonymy Resolution. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July 2017; pp. 1248–1259.
30. Speck, R.; Ngonga Ngomo, A.-C. Ensemble Learning for Named Entity Recognition. In Proceedings of the International Semantic Web Conference, Riva del Garda, Italy, 19–23 October 2014; Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 519–534.
31. Mackay, H.; Carne, C.; Beynon-Davies, P.; Tudhope, D. Reconfiguring the User: Using Rapid Application Development. *Soc. Stud. Sci.* **2000**, *30*, 737–757. [\[CrossRef\]](#)
32. Lindsay, C. From the shadows: Users as designers, producers, marketers, distributors, and technical support. In *How Users Matter: The Co-construction of Users and Technologies*; MIT Press: Cambridge, MA, USA, 2003; pp. 29–50.
33. Fleischmann, K.R. Do-it-yourself information technology: Role hybridization and the design-use interface. *J. Am. Soc. Inf. Sci. Technol.* **2006**, *57*, 87–95. [\[CrossRef\]](#)
34. Park, J.; Boland, R. Identifying a Dynamic Interaction Model: A View from the Designer-User Interactions. In Proceedings of the 18th International Conference on Engineering Design (ICED 11), Copenhagen, Denmark, 15–18 August 2011; pp. 426–432.
35. Karimzadeh, M.; Pezanowski, S.; Wallgrün, J.O.; MacEachren, A.M.; Wallgrün, J.O. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *TransGis* **2019**, *23*, 118–136. [\[CrossRef\]](#)
36. Ritter, F.E.; Baxter, G.D.; Churchill, E.F. User-Centered Systems Design: A Brief History. In *Foundations for Designing User-Centered Systems: What System Designers Need to Know about People*; Springer: London, UK, 2014; pp. 33–54.
37. Finkel, J.R.; Grenager, T.; Manning, C. Incorporating non-local information into information extraction systems by Gibbs sampling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, Michigan, 25–30 June 2005; pp. 363–370.
38. Cunningham, H. GATE, a general architecture for text engineering. *Comput. Hum.* **2002**, *36*, 223–254. [\[CrossRef\]](#)
39. Ratinov, L.; Roth, D. Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Boulder, Colorado, 4–5 June 2009; pp. 147–155.
40. Williamson, J. What Washington Means by Policy Reform. In *Latin American Adjustment: How Much Has Happened?* Institute for International Economics: Washington, DC, USA, 1990; pp. 1–10.
41. Buja, A.; Cook, D.; Scientist, D.F.S.R. Interactive High-Dimensional Data Visualization. *J. Comput. Graph. Stat.* **1996**, *5*, 78–99. [\[CrossRef\]](#)
42. GeoJSON FeatureCollection. Available online: <https://tools.ietf.org/html/rfc7946#page-12> (accessed on 1 January 2018).
43. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
44. Melo, F.; Martins, B. Automated Geocoding of Textual Documents: A Survey of Current Approaches. *TransGis* **2017**, *21*, 3–38. [\[CrossRef\]](#)
45. Kordjamshidi, P.; Van Otterlo, M.; Moens, M.-F. Spatial Role Labeling Annotation Scheme. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, 19–21 May 2010; pp. 413–420.
46. Wallgrün, J.O.; Klippel, A.; Karimzadeh, M. Towards Contextualized Models of Spatial Relations. In Proceedings of the 9th Workshop on Geographic Information Retrieval GIR '15, Paris, France, 26–27 November 2015; ACM: New York, NY, USA, 2015; pp. 14:1–14:2.

