

Article

Event Geoparser with Pseudo-Location Entity Identification and Numerical Argument Extraction Implementation and Evaluation in Indonesian News Domain

Agung Dewandaru ^{1,*}, Dwi Hendratmo Widyantoro ^{1,2} and Saiful Akbar ^{1,2}

¹ School of Electrical and Informatics Engineering, Bandung Institute of Technology, Bandung 40132, Indonesia; dwi@stei.itb.ac.id (D.H.W.); saiful@stei.itb.ac.id (S.A.)

² University Center of Excellence on Artificial Intelligence for Vision, Natural Language Processing & Big Data Analytics (U-CoE AI-VLB), Bandung Institute of Technology, Bandung 40132, Indonesia

* Correspondence: 33213033@students.itb.ac.id

Received: 10 August 2020; Accepted: 18 November 2020; Published: 28 November 2020

Abstract: Geoparser is a fundamental component of a Geographic Information Retrieval (GIR) geoparser, which performs toponym recognition, disambiguation, and geographic coordinate resolution from unstructured text domain. However, geoparsing of news articles which report several events across many place-mentions in the document are not yet adequately handled by regular geoparser, where the scope of resolution is either toponym-level or document-level. The capacity to detect multiple events and geolocate their true coordinates along with their numerical arguments is still missing from modern geoparsers, much less in Indonesian news corpora domain. We propose an event geoparser model with three stages of processing, which tightly integrates event extraction model into geoparsing and provides precise event-level resolution scope. The model casts the geotagging and event extraction as sequence labeling and uses LSTM-CRF inferencer equipped with features derived using Aggregated Topic Model from a large corpus to increase the generalizability. Throughout the proposed workflow and features, the geoparser is able to significantly improve the identification of pseudo-location entities, resulting in a 23.43% increase for weighted F1 score compared to baseline gazetteer and POS Tag features. As a side effect of event extraction, various numerical arguments are also extracted, and the output is easily projected to a rich choropleth map from a single news document.

Keywords: geoparser; geographic information retrieval; event extraction; argument extraction; information extraction; named entity recognition; conditional random function; lstm; semantic gazetteer; topic model

1. Introduction

The exponential rate of information shared through the world wide web provides ample opportunities to automate the understanding and extraction of information from the huge unstructured text collection. A lot of this information has embedded geographical references, either directly in forms of toponyms (place names entities) or indirectly via its references. One estimate stated at least 20 percent of Web pages include recognizable geographic identifiers [1] that are mainly present in unstructured form. It thus explains the development of numerous types of Geographical Information Retrieval (GIR) models, method, and prototypes with the aim of extracting, retrieving, and exploiting location and geospatial information within these unstructured textual data, such as online news articles [2], tweets [3], social media posts, or even blogs. These systems allow

improvement to useful types of applications ranging from analytics [4], health [5], retrieval [6], categorization, and many others by leveraging the geospatial data that is prevalent in the internet.

Unlike Geographical Information Systems, which process geospatial data from an already structured forms or records inside databases, GIR systems typically have to extract and infer geographic location or coordinates from many types of noisy information and ambiguities that are prevalent in the unstructured natural language form. Thus, a GIR system workflow typically starts with the geoparser component to extract geographic information from text, which is then followed by some indexing and retrieval mechanisms further down the pipeline. The regular geoparsing process within geoparser is composed of two subtasks [7]: (1) geotagging, i.e., detecting geographical references or toponyms from text, and (2) geocoding, which aims to resolve these into precise coordinates via some disambiguation method. The result will be further processed by GIR application to infer associations between varied information that is described in the document with the geographical coordinate of the resolved toponyms, which will be served or ranked across documents according to the geo-query input typically in some forms of thematic map.

A lot of efforts and iterations have been made in the field of geoparsing, from Woodruff, who introduced the first geoparsing prototype within GIPSY in 1994 [8], to Gritta's geoparser in 2019 [9]. However, the task of geoparsing is still an open problem to this date, due to the complex interaction between spatial, temporal, and thematic sub-space within text that needs to be addressed depending on the problem domain [10]. Indeed, geoparsers have been able to (1) infer geographic location from toponym mentions (which we called toponym-level resolution scope) or (2) infer single geographic focus of document (document-level resolution scope). Unfortunately, most of these geoparsers are still lacking the model and method to resolve coordinates at event-level resolution scope. This means that such geoparser is able to resolve precise location coordinates of (possibly) multiple events described within the document instead of only resolving or disambiguate coordinate of toponyms (toponym-level) or geographic focus of the document (document level). In terms of granularity, it sits between toponym-level geoparsers (such as [11–15]) and document-level resolution scope geoparsers (such as [6,16,17]).

We argue that the event-level resolution scope geoparser (or event geoparser for short) needs to be capable of (1) detecting what types of event(s) presented in the document and (2) infer the precise location of the event(s) reported (event geolocation) from the detected toponyms in the document. Additionally, (3) event geoparser should be able to discover which event argument(s) (especially numerical expressions/NUMEX) are associated with the detected event(s). This would enable richer, thematic geographic information retrieval usage such as spatial search, map visualization, and geospatial analysis from unstructured text input. In the bigger picture, the use of generated thematic map within GIR framework has been the motivation for this work, whose core component is arguably a type of event geoparser.

This paper presents a novel implementation of an event geoparser that is loosely based on ACE event model [18], which tightly integrates event extraction, and the toponym resolution, which is usually dealt with separately. The model decomposes an event into its trigger (or anchor), related entities, resolved (grounded) locations, and its semantic role arguments, especially numerical ones. The geoparser model cast the geotagging and event extraction as sequence labeling task; hence it uses state-of-art neural LSTM-CRF sequence labeling model as a statistical method employed on Indonesian news domain. For training purpose we constructed two set of corpora: (1) 645,679 editorially tagged news (i.e., with news keywords) documents of 13 years publication of Indonesian online news corpus with 107,133,817 words that were described in our earlier work [19] (which we will later identify as large corpus) and 83 news articles composed of 927 sentences annotated (disambiguated, geolocated, and event extraction tags with numerical arguments) sentences on four major geospatial events: flood, earthquake, fire, and accidents. This will be later identified as small corpus from which the event geoparser model is mainly trained. The geoparser also uses the smallest administrative level feature obtained from the resolved administrative level of the toponyms detected using Spatial Minimality Centroid Distance algorithm, which we derive from Leidner's Spatial Minimality algorithm [12]. This feature along with *event argument* feature proves to be very important

for the ability of the geoparser to detect the pseudo-location, which is necessary for geolocating events in the document.

To improve the model generalizability on unseen data, we also propose an exploratory model to learn semantic relatedness between topic label and its keywords from multi-labeled large corpus. This is called Aggregated Topic Model (ATM), which is trained from partitions of Labeled LDA [20] model output. The motivation of this model is to efficiently exploit a large number (in our corpus, reaching up to 44,280) of unique news tags as the labels offered by large corpus, which required too much RAM to process using Labeled LDA. We use ATM with Word2Vec to get list of keywords related to events and entities, which will be referenced as semantic gazetteer, adapted in the approach of [4]. The semantic gazetteer contains keywords that will be used to build handcrafted rules for event keywords feature or regular expression features to help improve geoparser's performance.

2. Related Works

2.1. Scope of Resolution of Geoparsers: Toponym-Level, Document-Level, and Event-Level

The majority of geoparsers work on either toponym-level resolution scope or document-level resolution scope. Toponym-level resolution scope means that it works with the goal that every toponym will have assigned coordinates, typically via some disambiguation and resolution process (grounding) from gazetteer references. This has been the most numerous type of geoparser and the most basic, in the sense that the output can be used to fetch the other resolution scope mode of geoparsers or possibly event coders. Examples of toponym-level geoparser are Edinburgh Geoparser [11], CLAVIN [21], and as component inside the GIR prototype of SPIRIT [6]. Leidner's Spatial Minimality algorithm [12] also works with this goal. On the other hand, geoparsers that have document-level resolution are set out to find the geographical focus of the document. The document-level scope resolution will resolve geographic grounding of document using some scoring based on the detected toponym, such as simplistic frequency of mention and distance from the beginning of document such as CLIFF [16] and Newstand [22]. More complex resolution involves scoring based on zone indexing as a function of topology of the toponyms such as part-of or adjacency relationship, as in Mahali [23]. These geoparsers offer both scopes of resolution, by doing document-level scope resolution after the toponym-level scope resolution. The output comparison of toponym-level resolution scope with event-level resolution can be seen in Figure 1.

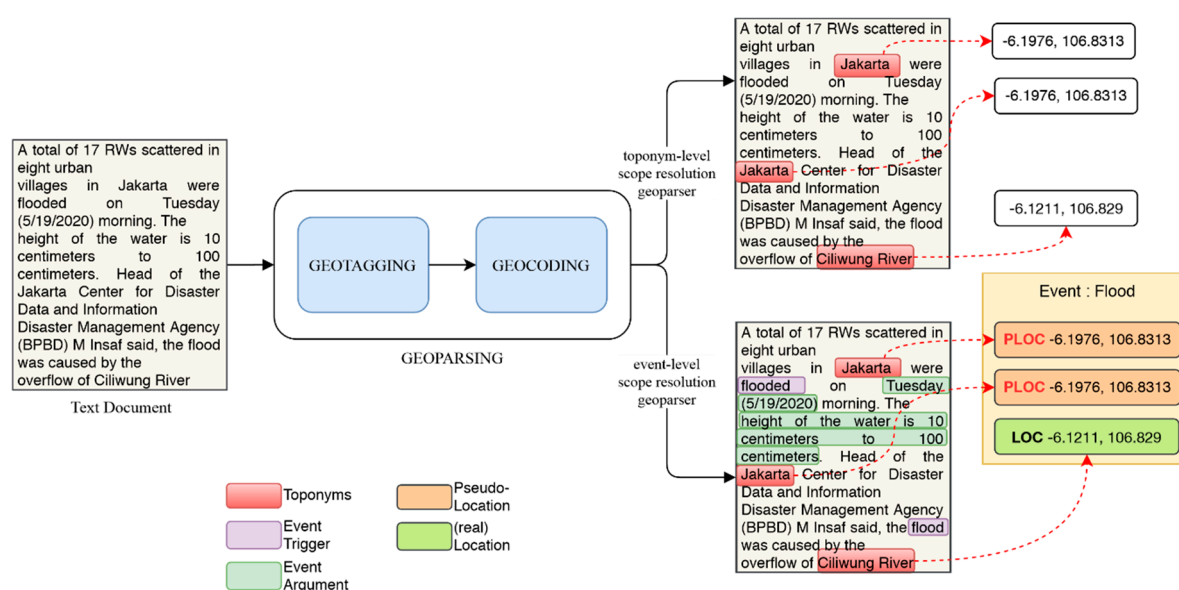


Figure 1. A comparison of toponym-level and event-level scope of resolution. Unlike toponym-level geoparser, the event geoparser is not only detecting toponym and resolving the coordinate but also detecting what event(s) happened and inferring which toponym is the real location (LOC) or only pseudo-location (PLOC) with regard to that event.

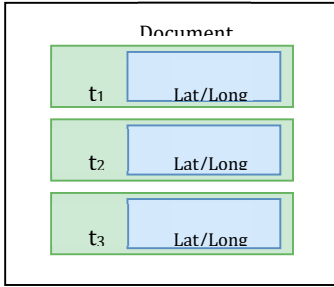
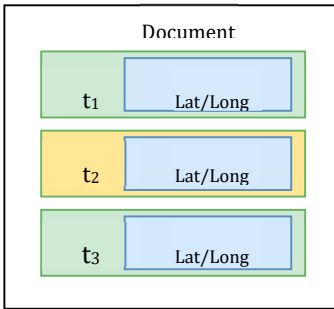
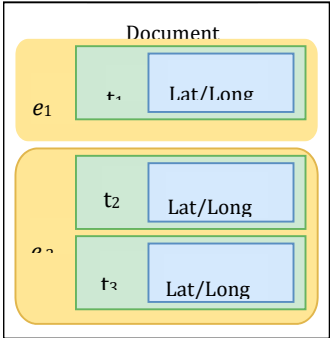
The last type—and the most recent development—is what we refer to as event-level resolution scope geoparser. It will try to detect event(s) within the document and to resolve the location or geographic scope of those events. There are only very few geoparsers which has this capability, and they are still very limited. Most geoparsers are only coupled or stacked with event coder (which sometimes targeting the output to certain event ontology codes such as CAMEO) to reach this capability. Typically, the approach is to start with toponym recognition (geotagging) using NER, following with event detection (or often referred as event coding) step; and ending with toponym resolution step using geoparser component, often as different independent module. This is the approach of the first prototype of event geoparser LocNZ [24] that is integrated within (as part of) InfoXtract architecture [25], TABARI parser [26] + Leetaru’s geocoder [27] in GDELT project, and PETRARCH + CLIFF [28]. ICEWS dataset was also prominent big dataset in similar area; however, its geoparsing description is rather not described adequately in [29]. Because of this independency of the geoparser module, the integration of event coding and geoparser is typically done in an opaque, black box approach: The geoparser does not know anything about the event structure or semantics; and the event coding system simply attaches the coordinate of the detected, resolved toponym to the location of the event. For example, CLIFF is document-level event geoparsing, and both Leetaru’s and LocNZ are toponym-level geoparsers. Hence, there is a gap between event and its location leading to inaccuracies of the toponym assigned, in other word, the toponym returned is not the real location (or irrelevant) of the event.

Mordecai [30] and Profile [31] are both of event geoparser which are capable of recognizing and resolve event location, so they have event-level resolution scope. Both operate within political event domain corpus. Profile uses an SVM-based classifier to differentiate between focus location entities with non-focus one. However, it works with a rather strong assumption that within document there is only one main event, hence, there is also only one geographic focus location of that event. This limitation makes Profile unable to handle a document which has more than one event or an event which has several locations, both of which are common within our corpus, and another dataset confirmed that such case is a common observation [28]. Mordecai is perhaps the only geoparser which explicitly defines event notion and performs linking of the (possibly several) event(s) with its locations. Mordecai models n token sentence as $X = \{w_1, w_2, \dots, w_n\}$. An event is symbolized as e_k and marked with anchor verb v_k (similar concept as *trigger* in ACE model) for the location of the event $G_k = \{g_1, g_2, \dots, g_j\}$. Each token has their event binary label $y_i^{(k)}$, either 1 or 0 depending whether w_i is the location toponym of that event k . The implication for this definition is quite significant. With this event paradigm, a document can be composed of more than one event, and each can have more than one location. However, even though Mordecai has the model of event (represented with its event anchor mentions) and the method to geolocate event, it does not model semantic role and its argument. Hence, the ability to detect event depends only on the features that the model uses, namely *part-of-speech* (POS) tags, pretrained GloVe [32], dependency label, and signed distance of word from the anchor [33]. Even though it is effective on the narrow, political dataset that Mordecai is trained upon, it may not be enough for broader domain. This motivated us to extend this model further to incorporate event arguments (and its *semantic role labels*) with ACE model as with notation described in joint event extraction model in [34]. It should also be noted that Mordecai does not use toponym resolution algorithm, leaving it vulnerable to toponym ambiguities.

These types and examples of geoparsers with respect of its resolution scope is listed in Table 1. In the next section, we will use and extend the Mordecai definition to include event arguments and resolved geographical scope.

Table 1. Types of resolution scope of geoparsers.

Type of Resolution Scope	Output Model Formulation and Illustration	
	$t = \text{toponym}$, $D = \text{set of words in the document}$, $\mathbb{E} = \text{set of events in the Document}$, $\mathbb{G} = \text{set of resolved coordinate/footprint}$	Example Geoparser/GIR

Toponym-level (geocoded toponym)	 <p>The diagram shows a box labeled 'Document' containing three green boxes. Each green box contains a toponym (t1, t2, t3) and a blue box labeled 'Lat/Long'.</p>	Output is geographical coordinate for each toponym in the document based on recognized toponyms entities within the document.	SPIRIT [6] Edinburgh Geoparser [11] Spatial Minimality [12] CLAVIN [21] Camcoder [15]
Document-Level (geographical scope of document)	 <p>The diagram shows a box labeled 'Document' containing three green boxes. Each green box contains a toponym (t1, t2, t3) and a blue box labeled 'Lat/Long'. The middle box (t2) is highlighted with a yellow border.</p>	Output is single geographic coordinate or scope of the document based on some scoring function of recognized toponyms within the document.	Web-a-Where[13] NewsStand [22] GeoTxt [14] Mahali [23] CLIFF [16]
Event-Level (geolocated event)	 <p>The diagram shows a box labeled 'Document' containing three green boxes. Each green box contains a toponym (t1, t2, t3) and a blue box labeled 'Lat/Long'. The first and third boxes (t1 and t3) are highlighted with a yellow border.</p>	Output is geographical coordinate for each locations of recognized event(s) within the document.	Mordecai [30] Profile [31] GDEL1 (TABARI + Leetaru [26]) Petrarch + Mordecai [30] Petrarch + CLIFF [28] InfoXtract [25] + LocNZ [24]

2.2. Mainstream Approaches in Geotagging: Gazetteer and Data-Driven NER Approach

The typical first task of a geoparser is to determine which tokens inside the text refer to names of a places. This process is commonly referred as geotagging or toponym recognition. Geotagging requires methods for discriminating location entities of place names (toponyms) from other entities. The dominant geotagging method used in most geoparsers is to incorporate gazetteer lookup, which is a lookup process from an external resource of place names and basic geographic information for simple string matching. Generally, the matching toponym string (which may consist of several tokens) inside the gazetteer indicates strong probability of such token being place names, with some exceptions needed to exclude highly ambiguous place names such as (city of) Reading, England. A gazetteer is a dictionary of place names or geographical thesaurus, often equipped with geospatial information (latitude and longitude or polygons) or extra information such as population size, administrative level, and alternative names. Gazetteers vary in their coverage of names, associated geographical information, and hierarchical structure. Common choice for gazetteer includes GeoNames, GNIS/GNS, WordNet, OpenStreetMap and GADM. A gazetteer can be classified regarding whether it has toponym hierarchy or not. Gazetteer which has toponym hierarchy is called ontological gazetteer [35]. We call an ontological gazetteer that maintains correct hierarchy for all its entries a strict gazetteer. GADM, for example, can be considered a strict gazetteer with four levels of administration from a total of 368,735 administrative areas. Geonames [36] is an ontological gazetteer with a much larger coverage, (totaling around 11.8000,000 features) although it does not have a strict geo-ontology. For example, there are many entries of a village (administrative level 4) that has been placed directly under a province level entry (level 1) whereas it should be under sub-district (level 3). The better the coverage, the better geoparser detect toponyms (related to recall performance). However, it must be noted that referential ambiguity (which is part of geo/geo ambiguities where

two or more toponyms share same name) is still a problem to be resolved, and the strict hierarchical information in gazetteer will also be useful for disambiguation strategy (the containment heuristic), which will be further discussed.

Toponym recognition can also be considered as a specialized form of Named Entity Recognition (NER) but with the focus on recognizing named geographical entities [12]. In the landscape of geoparsing, data driven NER approach is dominantly used along with gazetteer lookup, even though there are few *rule-based* geotagging approaches. For example, by detecting preposition such as “in” or “to” followed by toponym candidate such as Owen’s Kivrin [37]. Data driven approach requires an annotated corpus (often annotated using BIO scheme) which is typically trained to distinguish different entity types such as Person (PER), Location (LOC), or Organization (ORG). NER framework could use string matching of toponym from the gazetteer as one of its binary features, along with other feature such as POS tags [38], word forms, or capitalization. It means that not all matches will be considered as a toponym, depending on the classifier result. Using NER will generally be able to differentiate geo/non-geo ambiguities, and a lot of geoparsers are using external, specialized Named Entity Recognizer component for geotagging purpose to filter non-geographical names, such as MITIE (used in [30]), LingPipe (used in [23,39]), GATE ANNIE ([4,14]), Spacy [14], Stanford CoreNLP (used in [16,21]), NCRF++ (used in [9]), and others. Most of these NER in turn use statistical, data-driven sequence labeling model under the hood, such as Conditional Random Field (CRF) (CoreNLP and LingPipe), Maximum Entropy (Edinburgh Geoparser), or Hidden Markov Models [17].

Generally, both gazetteer and NER approaches have been successfully used by geoparsers to tag and extract the toponyms the text in the geotagging step. However, the main challenge here is that the extracted toponyms do not necessarily indicate the location of events mentioned in the news document. Furthermore, even though a toponym is indicating location (locative), it may not be precise enough to be stated as a location of a particular event. The reason for the problem and the taxonomy of toponyms with regard to event will be discussed in more detail in the next section.

2.3. Geotagging True (Locative and Precise) Location Toponyms Relative to an Event

The ongoing source problem of geotagging apparently stems from the inherent lexical ambiguities of toponyms and also syntax ambiguities of natural language. Therefore, it is important to analyze the taxonomy of toponyms. Gritta [7] divides taxonomy into literal and associative types. Literal toponym carries the notion of physical location. On the other hand, associative toponym is used in a context associated with the physical location (e.g., Mayor of Paris). While literal toponyms seems to be a major use case; it only comprises 53.5% in his evaluation on GeoWebNews corpus [7], with the rest of the uses being associative ones (46.5%). The similar structure dichotomy of toponyms is actually shared much earlier but from the toponym ambiguity standpoint. Amitay et.al. noted the ambiguities of toponyms present in the forms of geo/non-geo and geo/geo dichotomy [13]. The notion of geo/non-geo ambiguity refers to a toponym that has non-geographic disambiguation candidate(s) of the same name (such as Paris, France [GPE] vs. Paris Hilton [PER]). Similarly, geo/geo ambiguity appears when a toponym has more than one (literal) geographic referent of the same name (such as Paris, France [GPE] or Paris, New York [GPE]). This dichotomy has been followed and used in works of others such as [25,27,28] and [40,41]. However, for a geoparser to serve the event-level resolution scope discussed earlier, we argue that it still needs to discriminate further geographic, literal toponym mentions (geo/geo box in the Figure 2) with deeper dichotomy with regard to a particular event. This can be done using two criteria that needs to be satisfied: (1) event-locative (indicating location of event) and (2) precise (the location inference process prefers smaller areas than bigger ones). Thus, we will extend the dichotomy to focus on whether the toponym should be tagged as pseudo-location entities (PLOC) or real location (LOC) with respect to the detected event(s) in the document. In other words, even though geoparsers have been able and remove non-geographical toponyms (with regards to the first dichotomy), they still must identify which toponyms are locative and precise to which event (real location entities) and which toponyms are not (pseudo-location entities). As we soon discuss, this distinction is very important and has not yet been handled well by existing geoparsers.

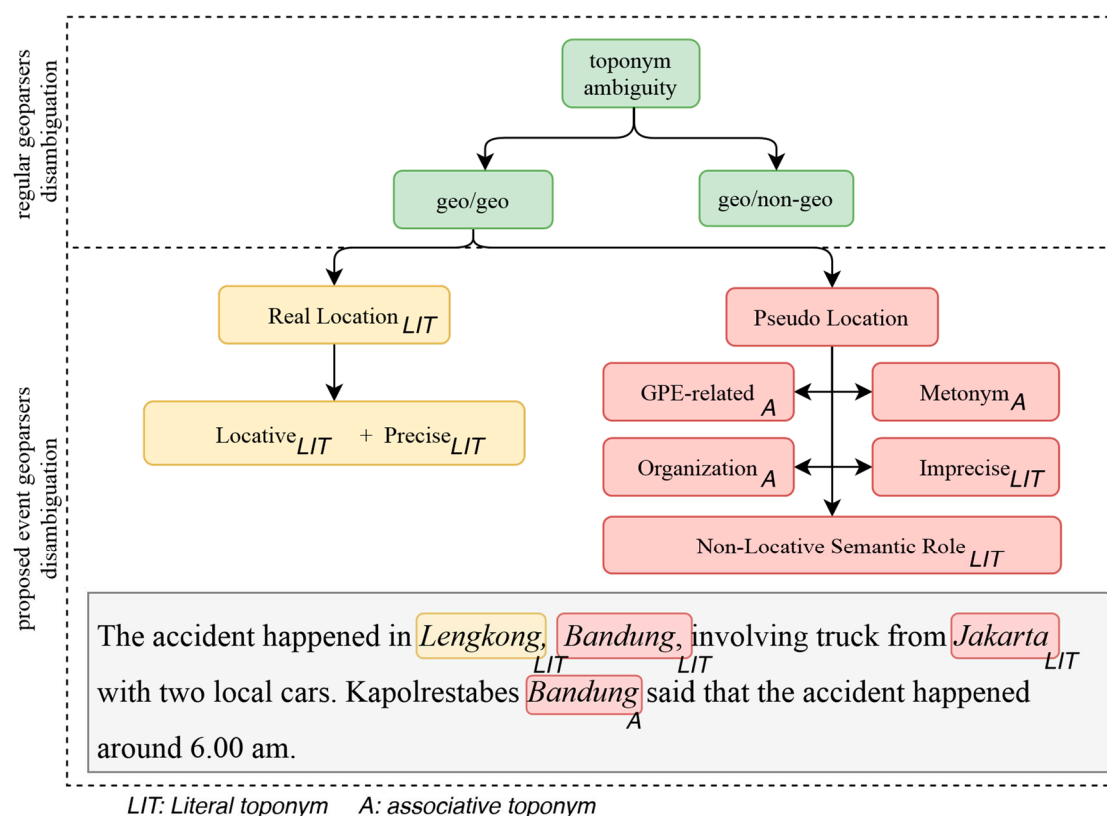


Figure 2. Taxonomy of toponym ambiguity. Even though regular geoparsers are already capable of filtering geo/non-geo ambiguities and assigning disambiguated coordinates to geo/geo referential ambiguities, they cannot yet handle event geolocation properly, i.e., recognize and resolve toponyms that are both locative and precise of *particular events* by discarding all “pseudo-location” entities which are irrelevant to that event. Note that pseudo-location entities may appear either as literal or associative toponyms as well.

The pseudo-location entities often occur in news corpora in the following associative, non-literal use cases of toponym: (1) as geopolitical entity modifier context, (2) metonymy [42], (3) as part of organization name. For example, in the sentence “U.S. President and North Korean leader hold a meeting in Singapore”, the United States and North Korea are both pseudo-location associative toponyms with regard to meeting event, because it appears in the geopolitical (GPE) context as a leader, not pointing to the location of the event (Singapore). Demonyms, the name of residents associated with such toponym, can be considered in this category as well. The second use case, associative metonymy, in this context is meant as a figurative, non-literal use of toponyms as a symbol of country or other entity. As an example, in this sentence, “Washington worked with Saddam before invasion of Kuwait”, where Washington represents United States as subject of the sentence, and hence, it is considered as a pseudo-location entity as well. Evidonym is where a toponym appears as a component in a multi-token toponym, often found as a part of organization name associated with some place [8,12]. Such as, “I studied at Massachusetts Institute of Technology”.

The pseudo-location entities may also appear in type of literal toponym usage, especially with (1) imprecise type mentions and (2) non-locative semantic role. Imprecise mentions are larger area toponym(s) which contain a more precise toponym. For example, in this sentence, “As result of the flooding, there were 128 residents in Balekambang, East Jakarta”. In Indonesia, East Jakarta is a city-level administrative area which contains Kelurahan (urban village) Balekambang as one of its indirect constituents. Recognizing a non-locative toponym (or for that matter, the inverse: a locative toponym) is not as straightforward as recognizing literal (non-associative) toponyms. Simply tagging toponyms based on lexical resources (e.g., gazetteer) is not enough (as in [6]) as toponym mentions in a single document do not always refer to where the event happened. These toponyms may appear in various sentence contexts in various syntactical patterns that present noises, which hinders the geoparser’s

performance. For example, toponym can be indicating literal but non-locative [7] with regard to an event: “The accident happened in Lengkong, Bandung, involving truck from Jakarta with two local cars”. The sentence illustrates an event (accident) that happen in Bandung, while the mention of Jakarta is obviously (to human reader) non-locative to that event (it is locative to the origin of the vehicle but not locative to accident event). We can say that literal toponym is not semantically equivalent to locative toponym. Locative toponym is always literal toponym, but the reverse is not always true. Thus, locative toponym set is a subset of literal toponym, which depends of a particular event type, which carries particular event semantics of a sentence.

Some may argue that discriminating literal from associative toponym using NER framework is sufficient for geoparsing, for example, the recent Gritta’s work on metonymy resolution [7]. However, it is clear in that sentence that all toponyms are literal. Lengkong, Bandung, and Jakarta are all literal toponyms. Hence, there is obviously a substantial need for discriminating the locative toponym. Moreover, it must be noted that NER methods do not offer coordinate-level accuracy or map-based disambiguation framework, which will be important for geotagging. Moreover, regular geoparsers or NER are not equipped with event semantics to differentiate locative vs. literal toponyms, as it is a necessary condition for the recognition.

The need of event semantics (such as matching event ontology, arguments, or type of events that may be inferred by a classifier as a particular label). Regular geoparsers (such as CLIFF-CLAVIN, Edinburgh Geoparser) are able to detect (tag) those toponyms without any issues. However, even though those toponyms are all literal toponyms (Jakarta, Bandung, and Lengkong), when it comes to the locative toponym question, “where do the accident event really take place?”, then, ideally, it will need to infer what event(s) has happened, the semantic role(s) and values that are associated with the event and, later on, correctly infer the real location entities where the event was located (i.e., finding locative and precise toponym) and its correct coordinates by geocoding technique.

Note that the precision of the reported event location within news articles may have various degrees depending on the event: It is quite common to pinpoint the location of an traffic accident to be very precise within a particular street, road segment, or coordinate, while an earthquake event may easily span across a province or even a country. This event-locative toponym is not solvable by NER only or geotagger as it may not have event-related semantics often produced by event extraction techniques. This event-related semantic can be provided by the event label and event arguments inferred by event and argument classification process, which will be explained in the next section.

2.4. Integrating Event Extraction Model into Geoparsing

Event extraction is a branch within information extraction field which has been initiated from 1980s and becomes more popular as big data and NLP technique matures [43]. Generally, the objective of event extraction is to have structured event information out from unstructured text. Some models of event emphasize on the temporal aspects and ordering structure of the events such as TimeML [44]. The TimeML model defines event anchor (event A happened at time T), event order (event A after event B), and event embedding (event A nested within event B). TimeML heavily models the temporal aspects of an event and less the spatial and grouping aspects of an event participant. Other event model like the 5W1H dated very early and is still being used to annotate the news corpus, such as the work of [45] and [46]. However, both models are not suitable to group various roles (especially the numerical arguments role which will be explained later) into the event structure.

Following Linguistic Data Consortium’s Automated Content Extraction (ACE) model definition [18] and [34], an event is defined as something that happens that relate to one or more arguments (participants, place, time, etc.) In this work, we are interested more in custom ontology assumptions to model the events. Therefore, we chose to base on the ACE model loosely, which is very flexible and has been used extensively in many domains. We do not have to follow the event types and subtypes definition, but it can be customized according to the domain needs and its ontology. The similar geographic information retrieval that uses ontology for extraction is the hazard related extraction [4]. The hazard ontology is used with a list of keywords called semantic gazetteer to

geolocate events. Unlike the machine learning approach here, it uses rule-based JAPE language (GATE) and does not extract various event argument slots except fixed spatial, temporal, and semantic keyword entities.

The majority of the geoparsers are using Named Entity Recognition (NER) technique to perform toponym recognition and then proceed with the disambiguation or retrieval without emphasizing the event semantics and its extraction. One of the implications of event extraction, especially the ACE model, is the possibility to extract (often numerical valued) arguments within the document, as in [34]. For example, in the sentence, “The explosion killed 7 and injured 20”, not only explosion events are recognized but also the quantity related to it (i.e., 7 person and 20 person). Another example is a typical accident event which has semantic roles such as location, the number of death victims, and the origin of the vehicle (the ontology of such event is presented on Figure 3). Within the context of geoparsing, the extracted event types and their arguments may provide additional information context to the event geolocation process for better inference, while the extracted arguments may be useful to provide a richer data for the generation of the thematic map. As far as we know, the integration of event extraction methods within geoparsing (or vice versa) is still very shallow or even lacking. Event extraction methods does not discuss coordinate-level accuracy while geoparsers aims for such accuracy but without knowing the event context of the toponym mentions. The integration of event extraction system and geoparsing is done typically by two separate stages where toponym-level geoparser works with raw text (without information of any event structure), and the output is attached to the event extraction result.

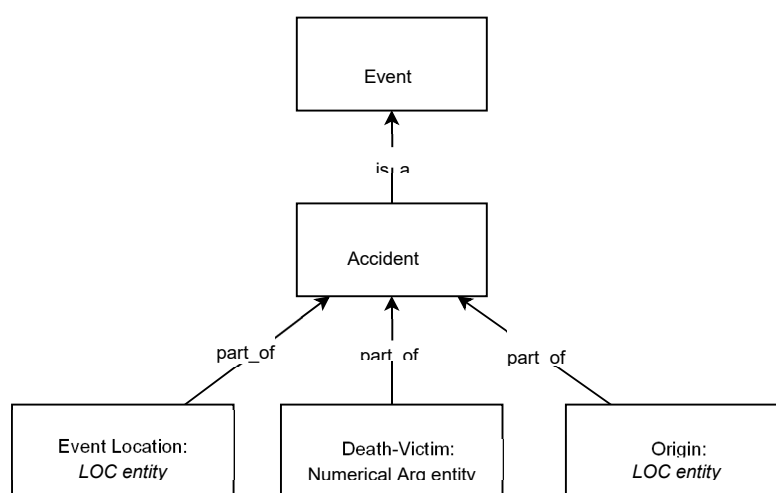


Figure 3. Sample Ontology of Vehicle Accident. Events can be modeled as grouping of various semantic roles and their arguments, forming templates for different types of events. For example, the Accident event has two semantic roles regarding location entity and one numerical argument.

Thus, the objectives we believe are still missing in state-of-the-art geoparser field are twofold: (1) the deeper integration of an event extraction framework for event geolocation method to resolve event-level resolution scope. This is to infer what type of event(s) are described in the document and in which precise location such event happened. Event extraction framework will provide event labels and event arguments which will provide richer semantic event context (in which the inferred location data is associated), which eventually will improve the performance of the geoparser; the numerical arguments extracted along with the event will provide a basis for automatic choropleth thematic map visualization that was noted in [19]. Another important implication of event-level resolution scope is that many of the location entities scattered through the text may not be relevant to the event at all. Thus, the second objective is (2) recognize the most relevant, precise toponym to the event. For these purposes, this paper introduces a novel geoparser type, which embraces event extraction framework with a special classifier to recognize *pseudo-location entities* to define valid location entities (toponym) but nevertheless irrelevant to the event, or such toponym may be relevant but not precise enough to the event entities inferred. The main contribution of this work is an event geoparser model which

integrates event extraction framework with geoparsing technique to locate event-level resolution scope of the document. The proposed model is equipped with pseudo-location identification method to further separate pseudo-locations from real locations, which improves the toponym resolution process.

2.5. Geocoding (Toponym Resolution) Process and Strategies

This section discusses toponym resolution step, which typically starts after toponym recognition. Toponym resolution sought to resolve referential ambiguities the literal toponym detected in toponym recognition. For example, given the toponyms in a document {Paris, France, Eiffel}, which location of Paris is the correct referent? Is it (a) Paris, France; (b) Paris, Maine, United States; or one of many other Paris from tens of possible candidates around the world? To answer this question, typically the researcher employs a set of toponym resolution heuristics. These heuristics generally represent toponym resolution insights that are coded into the system as simple rules or simplifying assumptions. For example, the *population heuristics* prefers higher population referent to lower population referent candidates [47]. Thus, if ambiguous places are present, the system will resolve it or prefer the most populated place. This is used in the work of [11,13], and many others. Other heuristic that is often used is *one geographical scope per document*. This rule confines so that there is only one focal geographical point within the document [16]. Similar to that, there is also the very common “*one sense per discourse*” heuristic, which assigns only one interpretation across several instances of the same toponym, used in [13,48]. Another heuristic used in the context of document-level geoparser is that of *frequency heuristic*: The geoparser prefers the interpretation whose number of occurrences of the toponyms is the highest within the document. The more it appears, the more likely a geographic entity candidate becomes a winner for representing the focus of the document [16]. These heuristics are often used as a component in a larger data-driven method such as clustering approach [49] or classification method [50]. The hierarchical knowledge embedded in gazetteer is often used to help the disambiguation [13,17], in which parent toponym appearance would increase the likelihood of the child toponym and vice versa. This is referred to as *containment heuristic* (or *local context heuristic* if it happens within a short window of text) and will be discussed more in the next section.

Lastly, toponym resolution strategies often make use of the map information available to prefer lesser place distance or *geographic proximity* [50] or overlapping areas [8,51]. In these systems, the further the place from geographically calculated averaged centroid, the lesser importance it will be given. Typically, this will need ontological information within gazetteer. Similar strategies found in [52] and [47]. This strategy is introduced in [53] and pseudo coded in [12] as a part of the baseline algorithm. This geographic distance strategy is also used in [17] which evaluates the distance between all possible toponym candidate pairs. Another similar algorithm that is often used in other works called the *spatial minimality*, based on the premise (called *geometric minimality heuristic*) that the correct place candidates compose the smallest region that is able to contain the whole set of toponyms inside a document [12].

Generally, these heuristics depend on the information of the geographic coordinate and taxonomy within gazetteer and method to evaluate area or distance between points. In this event geoparser work, we are implementing only a sufficient subset of these explained heuristics, namely, the one sense per discourse, the geometric minimality, and geographic proximity heuristics, to perform toponym resolution that will be used further in event resolution stage.

2.6. Increasing Model Generalizability with Topic Modeling

Enumerating all possible events semantic within a large corpus can be done by constructing *semantic gazetteer*, which is a list of keywords that can be used to represent concepts such as in [54]. This keyword, if obtained from a large corpus, will be able to increase the performance of the model on unseen data. However, the manually constructed keywords process would be time consuming and biased; thus, it gives a motivation for some automated method to help this exploration process. Machine learning approach to detect event triggers has been done, for example, by [55]. Topic

modeling is often used as an automatic statistical method of dimensionality reduction for clustering articles into a set of topics [56], which itself is a distribution of related keywords. Thus, topic modeling is a good approach to the semantic relatedness concept [57]. The output of topic model is typically a set of topic clusters, each of which is essentially probability distribution over words. This would provide a cluster of related terms, which resembles the notion of topic relations between words. The “top-words” are a collection of most related words that constitutes a particular topic; thus, we can use it as a feature for classification for event extraction, for example, by supplying binary features for context words, in order to detect the existence of event trigger words.

Topic modeling models are typically trained in unsupervised learning fashion, with the main input being the number of topics that they should produce from the training session. For example, in the Latent Dirichlet Allocation (LDA) model, the main parameter is K (number of) topics [58]. This is excluding the hyperparameters α and β that can be further fine-tuned. However, most of the corpus within the news domain has some categories and tags, and the LDA model does not make use of tags within the document as a guide for its clustering of topics. This is a disadvantage because most news publications have document “tags” (or “labels”, loosely speaking, not to be confused with dataset label) that works as a topic, for example, an article about particular flood can have “flood”, “disaster”, and some tags indicating city location as well (such as “Jakarta”). These tags are valuable and can be used as additional supervision for the LDA, providing a multi-label learning that is explored by many authors [20,59,60]. With the introduction of tags as label, the unsupervised nature of LDA becomes supervised in Labeled LDA.

One of the LDA derived models that use document tags is the Labeled LDA [20], which puts a one-to-one correspondence constraint between document tag and latent topic. A topic has a string label (caption) taken from a document tag that can be used for further inference. Unlike the unsupervised LDA, Labeled LDA (LLDA) incorporates supervision with the above mechanism; hence, there is no need to specify K as it is determined by the number of the unique tags in the corpus. This solves the problem of specifying K by trials, as is often the case in topic modeling frameworks: There is no clear-cut method to specify the number of clusters of the topics [61]. However, LLDA consumes a lot of RAM as the number of tags increases, such that typical RAM may not be sufficient for extreme labeling (more than 10,000 unique tags).

The work presented in section 4.3 introduces Aggregate Topic Model (ATM) to help the event geoparser learn the semantic relatedness of terms and event structure based on document tags within the large corpus. ATM discovers topic words and its tag labels by doing sufficient partitioning and training each partition using LLDA. Using the model, the training can be done in smaller chunks of dataset; hence, the RAM consumption is much less and is able to handle tens of thousands of tags. This aggregated topic model will be used to construct semantic gazetteer along with word2vec unsupervised word embedding model [62] to assist the widely used conditional random field (CRF) sequence labeler to provide better precision and recall of the event trigger classification.

3. Geospatial News Event Extraction Corpus

The objective of this corpus is to be the material of experiment from which we can gain improvement by integrating event extraction framework into the geoparsing. To the best of our knowledge, there is not yet any news corpus that provides both the correct geographical disambiguation as well as event extraction labels and that is suited to training and testing, much less one in Bahasa Indonesia. The criteria that we looked for in the news dataset was (1) that it covered major geospatial events (2) that it resolved all place names to the correct coordinate and administrative entities, and (3) that it had event-semantics in form of annotations which emphasize on numerical arguments of certain semantic roles slots within an event. For example, MUC corpus is one of the first event extraction corpus. The ACE 2005 corpus has explicit event structure and coreference task. However, it has very few numerical (NUMBER or NUMEX) argument slots, and it is not toponym disambiguated nor geoparsed/grounded to a coordinate level. TR-CONLL [63], Wiktor, and GeoWebNews [64] provided geoparsed corpus, but they did not provide any event extraction annotations, let alone numerical arguments. The spatiotemporal and thematic corpus of

Wang [4] has event semantic textual information (non-numerical) and geoparsed from 50 CNN news report about hazard; unfortunately, it is not an open dataset, and we are not able to access it. In the Indonesian context, there is the 5W1H-style news extraction and corpus [46] but without geoparsed toponyms and detailed event semantics. These circumstances motivated us to contribute one in Bahasa Indonesia.

We first used the corpus of our earlier work [19], which consisted of 13 years of news articles (2005–2018), totaling 645,679 documents with 109,279,585 words and around 150,000 unique tokens from Indonesian online news site *detik.com*. This corpus (which will be referred as 650K documents corpus or large corpus) can be seen as a multilabel classification corpus, with document tags treated as labels. There are 44,280 unique document tags, with an average of around 2 labels per document. All of these articles are in Bahasa Indonesia (Indonesian formal language); however, the toponyms mentioned are often international as is (for example when referring to fire in California) or reference adaptation of Bahasa Indonesia. This corpus follows Zip’s Law with a slope close to -1 , as with many other corpora in other languages [65], indicating the similar basic usage distribution pattern of our corpus (see Figure 4).

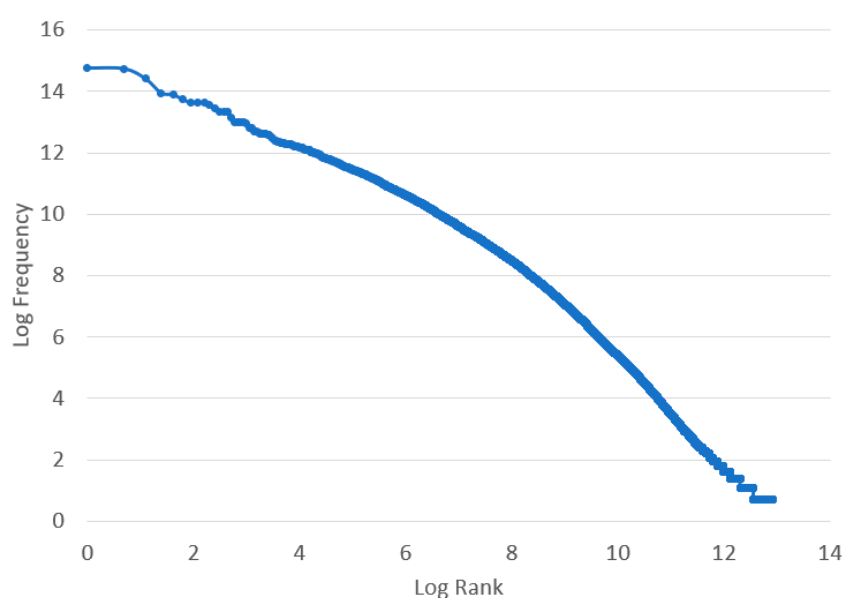


Figure 4. Zipf curves for the Indonesian corpus (650K).

Secondly, we selected a random subset of the corpus of the four most mentioned geospatial events according to Aggregated Topic Model count of topic suggestions: (1) flood (*banjir*), (2) quake (*gempa*), (3) fire (*kebakaran*), and (4) accident (*kecelakaan*). An ontology similar to Figure 3 for each of these events is developed to guide the annotation. It is important to note that the model of event should permit multiple instances of event at multiple locations within single news story. We use four annotators to work with 927 sentences from 83 articles from the subset corpus from *detik.com*, *kompas.com*, and *cnnindonesia.com*. The annotations are done for each token following the BIOES-style annotation tagging format. The tags are organized into the following tags code (Table 2). This smaller set of corpora (which will be referred as small corpus or event geoparsing corpus) contains part-of-speech tags, entity types annotation, event annotation, geospatial disambiguation annotation, and pseudo-location tags, which are obtained from InaNLP tagger.

In Bahasa Indonesia, the morphological derivation that modifies noun to adjective applicable to a toponym is not known. For example, in the sentence, “Saya warga Indonesia yang tinggal di Indonesia” (I am Indonesian citizen who lives in Indonesia), the first instance of word Indonesia is seen as an adjective that modifies the noun “warga” (citizen), constructing a demonym (noun) or adjective people related to a place). Notice that there are no morphological differences between the two-word forms (morphemes), unlike in English, which uses the *-ian* suffix (i.e., Indonesia vs. Indonesian). In the annotated corpus however, the POS tag (output from InaNLP) does not yet

differentiate between the two and simply labels them as NNP (proper noun). This posed a challenge for the pseudo-location identification task as it has to differentiate locative toponyms (which should be present as NNP instead of JJ/adjective).

The entity annotation tags contain labels of event triggers (EVE), event arguments (ARG), organization (ORG), and locations (LOC). Typical (NER) Person (PER) label is not used because a lot of this information is already represented by the argument entity (e.g., OfficerOfficial-Arg) in our corpus. The second annotation is that of Event triggers subtypes. Each of the events is further annotated into either four main event tag codes (Fire, Accident, Quake, and Flood) or secondary event codes that will not be included in our evaluation (Rain, Jam, Landslide, Meeting, and Evacuate).

Table 2. Entity tags description.

Tag	Description and Examples
EVE	Description: Event Triggers: word(s) that indicate an event has occurred. Examples:
	1. Flood happened in... (Banjir terjadi di...) 2.that the fireworks from the band triggered fast-moving fire flame . (... bahwa kembang api dari band, memicu kobaran api yang bergerak cepat.)
ARG	Description: Non-named arguments related to event. May include numerical or non-numerical arguments. I. Event Arguments for Flood 1. <u>Height of flood: Height-Arg</u> The height of the water reached 2 m ... (Ketinggian air yang mencapai 2 m ...) 2. <u>Number of Victim (Deaths): DeathVictim-Arg</u> At least 41 people killed due to the flood. (Sedikitnya 41 orang tewas akibat banjir ini.) 3. <u>Number of Evacuee: Evacuee-Arg</u> Indonesian Field Hospital handled 9 victims and 346 evacuees . (Rumah Sakit Indonesia di Nepal Tangani 9 Korban dan Tampung 346 Pengungsi) 4. <u>Number of Affected houses: AffectedHouse-Arg</u> Flooding caused 4991 houses to be submerged... (Banjir menyebabkan sekitar 4.991 rumah terendam...)
	II. Event Arguments for Quake: 1. <u>Magnitude (Richter or MMI unit): Strength-Arg</u> A 5.2 Richter earthquake shakes Maluku waters. (Gempa 5,2 SR Goyang Perairan Maluku) 2. <u>Quake Center: Central-Arg</u> The coordinates of the earthquake are -3.4 Latitude 128.41 Longitude ... (Titik koordinat gempa ada di 3.4 Lintang Selatan dan 128,41 Bujur Timur) 3. <u>Quake Depth: Depth-Arg</u> The depth of the earthquake was 10 km . (Kedalaman gempa 10 km)
	III. Event Arguments for Fire: 1. <u>Number of houses burnt: HouseBurnt-Arg</u> A house at Mampang Prapatan burned... (Sebuah rumah di Mampang Prapatan Terbakar) 2. <u>How many fire hotspots: Point-Arg</u> There are nine fire spots ... (ada sembilan titik api ...)

	<p>3. <u>Units of fire truck dispatched: DispatchedTrucks-Arg</u> ...12 firetrucks were dispatched. (...12 damkar dikerahkan)</p>
	<p>IV. Event Arguments for Accident:</p> <p>1. <u>License plates: Plate-Arg</u> ... which has B 9667 ZX license plate. (...beropol B 9667 ZX)</p> <p>2. <u>Vehicle type involved: Vehicle-Arg</u> ...hit hard on the back of the truck. (menghantam keras bagian belakang truk)</p> <p>3. <u>The length of jam: Length-Arg</u> The accident caused up to 2 km traffic jam. (kecelakaan mengakibatkan kepadatan kendaraan hingga 2 km)</p> <p>4. <u>Origin or Destination: FromTo-Arg</u> The accident caused up to 2 km traffic jam. (kecelakaan mengakibatkan kepadatan kendaraan hingga 2 km)</p>
	<p>V. Others (may appear in more than one events above):</p> <p>1. <u>Numerical Monetary loss: MonetaryLoss-Arg</u> The loss is estimated around hundreds of millions of rupiahs. (Kerugian diperkirakan mencapai ratusan juta rupiah.)</p> <p>2. <u>Time or Date of event: Time-Arg</u> -...at 15.25 WIB. (...pukul 15.25 WIB). -... as reported by AFP news agency on Friday (3/25/2011) (...dilansir kantor berita AFP, Jumat (25/3/2011))</p> <p>3. <u>Cause of event: Cause-Arg</u> Example:...caused by River Kuncir overflow. (...disebabkan luapan air Sungai Kuncir)</p> <p>4. <u>Affected families: AffectedFamily-Arg</u> ...which caused 935 families to be affected. (...menyebabkan 935 KK terdampak)</p> <p>5. <u>Street names: Street-Arg</u> There is a fire near Street KS Tubun Raya behind the Bimo Hotel (ada kebakaran di dekat Jl KS Tubun Raya belakang hotel Bimo)</p>
ORG	<p>Organization (such as military or civilians) Rank/Positions within it (<i>pangkat/jabatan</i>) Useful for classifying Pseudo LOCs. Example: BPBD and TNI... (BPBD bersama TNI...) Governor of East Java... (Gubernur Jawa Timur...)</p>
LOC	<p>Location or Toponym in types of GPE (geopolitical entities) administrative unit. Ranging from lowest administrative level to highest (Village, Sub District, Municipalities/Cities, Province, Country). Pseudo Location will be labeled as PLOC. Example: The flood again submerged 11 villages in Gandusari Subdistrict, Trenggalek Regency. (Banjir kembali merendam 11 desa di kecamatan Gandusari Kabupaten Trenggalek.)</p>
O	Other Entities

The next set of annotations are the argument types for each relevant event. We are following the ACE approach by defining subtypes of Events and Arguments tags. This provides the event codes and semantic contexts of each argument (see Table 2, ARG row).

The next two annotation sets focused on the geographical aspects. We disambiguated (geocode) each of the LOC entities manually and also provided the list of disambiguation options along with the approximate central coordinate (centroid) of that geographical feature. Most of these LOC tags are in the form of Geo-Political Entities (GPE) definition of ACE, so it is desirable to use an administrative-based gazetteer to reference them. Moreover, there appears to be a recurring pattern of specifying toponyms in a consecutive and hierarchical manner, starting from the lower level to the higher level (e.g., from village, up to the province level).

Among the open data gazetteers that are available for use are Open Street Map (OSM), GADM, and Geonames. GADM provides a very close coverage of GPE administrative taxonomies. It divides the world into 5 administrative levels: country (Level 0), provincial (Level 1), municipalities (Level 2), sub-district (Level 3), and village (Level 4). Even though the total entries or coverage are not as comprehensive as Geonames, it is more rigorously structured in the sense that every upper administrative area is always composed of smaller elements. This is in accordance with the containment heuristic that we have discussed earlier and will then be used in smallest administrative level feature discussed in Section 4.2. Geonames also has hierarchical information, but there are gaps in many entries. For example, a sub-district named Madiun is listed as direct child of East Java province, whereas it should be listed under a regency before province. OSM excels on specifying the street-level toponyms; however, in the context of the visualization of large-scale geospatial event, we felt this advantage is too fine-grained.

In light of these advantages, we choose to use GADM as the main reference for the location coordinates annotations and for the geoparsing later on. However, GADM does not provide a centroid for parent nodes, so we calculated them on the basis of the average latitude and longitude of all centroids under the node and put it next to the location tagged tokens. We initially used BRAT tool to annotate the corpus; later, it was converted to a plain text representation manually.

The last part of the corpus construction is the discussion of pseudo-location entities (PLOC) definition, which is an important label component in the annotation. In the corpus, we assigned pseudo-location entities to be precise location entities (toponym) which are inhabited place names and a GPE which is locative and precise as explained in the introduction section. For an article document offering more fine-grained toponyms for an event (smaller area location), this will be normally selected compared to bigger area. This is a sensible heuristic for many events such as Flood, Accident, and Fire. Particular exception was made with regard to a huge area-related events such as Earthquakes, where it is possible to be affected across large administrative areas such as provinces or even countries.

The second locative reference criteria meant to discriminate real geographic location attribute with associative references. For example, in this sentence: “USGS (United States Geological survey) stated that the quake situated in area around 68 km to the west of Namche Bazar, near Mt. Everest”, the United States is a valid toponym but only associative. It clearly does not refer to a locational attribute of the quake event (pseudo-location). Hence, it would be labeled with PLOC, while Namche Bazar would be labeled as LOC. Mt. Everest is not labeled as LOC as we do not consider it as administrative region. Instead, uninhabited places or geographical landmarks are typically labeled as ARG label with proper semantic roles attached.

The small corpus is named Event Geoparsing Indonesian News Dataset and has been published in *IEEE Dataport* [66] with the following label statistics of entities, events, and arguments (Table 3)

Table 3. Label statistics within Event Geoparsing Indonesian News Dataset. For brevity, the B- and I-prefix variation for each tag are collapsed into one label category.

Type	Label	Count	Type	Label	Count
Entity	LOC	454	Argument	Duration-Arg	31
	PLOC	627		AffectedVehicle-Arg	23
	EVE	700		AffectedFacility-Arg	8
	ARG	2016		AffectedField-Arg	10
	ORG	571		AffectedPeople-Arg	15

Event	ACCIDENT-EVENT	131	AffectedVillage-Arg	35
	FLOOD-EVENT	207	AffectedRT-Arg	4
	QUAKE-EVENT	128	AffectedFacility-Arg	8
	FIRE-EVENT	180	Time-Arg	422
	LANDSLIDE-EVENT	18	Published-Arg	79
	MEETING-EVENT	4	Reporter-Arg	39
	JAM-EVENT	16	Evacuee-Arg	20
Argument	Vehicle-Arg	121	Spot-Arg	9
	Hospital-Arg	50	DeathVictim-Arg	201
	Place-Arg	1070	WoundVictim-Arg	29
	Street-Arg	159	MonetaryLoss-Arg	3
	Cause-Arg	52	OfficerOfficial-Arg	408
	FromTo-Arg	57	Depth-Arg	28
	Plate-Arg	70	Central-Arg	89

4. Approach

This section describes the approach, the design, and the implementation of the proposed event geoparser prototype. It will be started with the formulation (Section 4.1) and followed by the architectural view, which explains the stages of the geoparsing (Section 4.2). These are the key concepts and essential for event geoparser model summarized in Figure 5. The next discussion of this section is to improve the generalizability of the model by doing semantic exploration to derive the semantic gazetteer (purple box on the figure) using a supervised topic model for news corpus that has multiple tags for each of the articles (Section 4.3–4.4). This section will be finished with the discussion of Spatial Minimality algorithm improvement in order to disambiguate toponyms on degenerate polygon cases (Section 4.5). Note that toponym disambiguation is located on Step 2 (Geocoding/Toponym Resolution on Figure 5).

4.1. Task Formulation

As noted in Section 2, we are going to use and extend the definition from Mordecai to further include several additional variables in the model. First, we reiterate the model of a sentence, which is composed of n tokens, $X = \{w_1, w_2, \dots, w_n\}$. The binary-valued variable $y_i^{(k)}$ which shows the location toponym of an event is now supplanted by n -ary label output variables, a , t , r , p , with the following definitions, related to word w_i :

$$a_i^{(k)} = \begin{cases} q & \text{if } w_i \text{ is the token that has entity type } A_q \text{ for event } k \\ 0 & \text{if otherwise} \end{cases} \quad (1)$$

where A_q is a q -th element from set of all entities types, $A = \{A_1, A_2, \dots, A_n\}$. Entities types comprised of event trigger entities (“B-EVE” and “I-EVE”), organization entities (“B-ORG” and “I-ORG”), arguments (“B-ARG”, “I-ARG”), and locations (“B-LOC”, “I-LOC”). Note that we are using BIO notation in entity labels so the B/I prefix applied to each type indicates its position at the beginning of entities or inside them. Similarly, the event trigger type (t) and semantic role label type (r) each is expressed as

$$t_i^{(k)} = \begin{cases} q & \text{if } w_i \text{ is an event trigger entity that has event trigger type } T_q \text{ for event } k \\ 0 & \text{if otherwise} \end{cases} \quad (2)$$

$$r_i^{(k)} = \begin{cases} q & \text{if } w_i \text{ is an argument entity that has semantic role type } R_q \text{ for event } k \\ 0 & \text{if otherwise} \end{cases} \quad (3)$$

where T is set of all event trigger labels (also prefixed with BIO codes) such as “B-FLOOD-EVENT”, “I-QUAKE-EVENT” and R is set of all semantic role labels like “B-Height-Arg”, “I-DeathVictim-Arg”, etc. (Please refer to Table 3 for all possible labels for semantic roles and event types). Next, we introduce an important variable for identifying the event geolocation. the pseudo-location labels

which subcategorize LOC entities into either pseudo-location (PLOC) or real location (LOC) each also prefixed BIO scheme:

$$p_i^{(k)} = \begin{cases} q & \text{if } w_i \text{ is a location entity that pseudo-location type } P_q \text{ for event } k \\ 0 & \text{if otherwise} \end{cases} \quad (4)$$

Note that we do not limit verb word type as anchor word. Instead, it may be single multi-word non-verb entities that are deemed relevant [67].

Last but not least, is the $g^{(k)}$ variable, which denotes the resolved geographic location entities (toponym) for an event k . Unlike the variables explained before, it does not represent sequence labels in the document. Instead it represents the geographic coordinate of true location(s) of the event; hence, the domain is geographic. In many news articles, it is possible that an event has several true locations, e.g., quake event can easily span multiple places or cities reported. Thus, the set of true location(s) are obtained by the process of resolving toponyms (geocoding) of the remaining location entities after discarding the pseudo-location entities associated with the event.

These sets of variables a, t, r, p, g will then need to be linked with event e using the index k denoted as superscript; hence, the event location of $e^{(k)}$ is indicated by $g^{(k)}$, and its related arguments can be seen by examining $r_i^{(k)}$ and so forth. In the case where there are more than one event instances of the same type found within an instance, it is likely that it needs to be co-referenced together. However, the topic of event coreference resolution is not our focus in this work as the strategies may vary for different domains, independent of the topic of event geoparsing.

4.2. Three Stages of Event Geoparsing Workflow

This section will describe our architectural, systematic approach for integrating geoparsing with event extraction to provide event-level resolution scope, which we like to refer as event geoparsing. We will first define the regular pipeline of geoparsing and describe the additional pipeline where the event extraction process takes place. We extended the regular workflow of GIR and geoparsing process following [64] and generalized from our discussion from an earlier section, by combining regular geoparsing stage with event extraction stage, and concluded with event-level scope resolution stage.

In total, there are six steps grouped into three stages which are briefly discussed as follows. The first stage is the standard *toponym-level geoparsing stage*, which is comprised of the following steps:

1. Geotagging, in which named literal geographical entities (toponyms) are recognized from other named entities. This is where the Named Entity Recognition is typically invoked to recognize location entities.
2. Geocoding (or toponym resolution step) in which correct toponyms are disambiguated from other toponym candidates (potential referents) and then assigned correct geographic coordinate. This is obviously a toponym-level scope resolution and calculated using spatial minimality based algorithm.

We are hoping to have a deeper integration of event extraction into geoparsing by extending those original two steps, in a more transparent flow of features unlike the typical combination of event coder + geoparser such as or TABARI/Leetaru or PETRARCH/CLIFF geoparser [68]. In particular, the model runs event extraction stage after the geoparsing stage (geotagging and geocoding), followed by event level scope resolution stage, as can be seen in dotted boxes in Figure 5. This will provide event record data to be stored along with place data. The second stage is the *event extraction stage*, which comprises two steps:

3. Event trigger classification. This step is to recognize the event triggers and provide event code label based on the detected class.
4. Argument Extraction. This step is to recognize semantic roles within event and extract arguments, including numerical ones.

The final stage is to resolve the location of the event (*event-level geoparsing*). This stage is comprised of the following steps:

5. Pseudo-location Identification. This step is to classify each LOC entities detected in the step 1 into either PLOC (pseudo-location) or LOC (real location).
6. Event coreference resolution. This step is to group several events of the same instance in the document into a single event structure.

The entire process can be seen in the diagram on Figure 5, which will be described in more detail as follows for each stage. The *geoparsing stage* starts with geotagging step, which involves cleaning, sentence splitting, and tokenization of the small corpus.

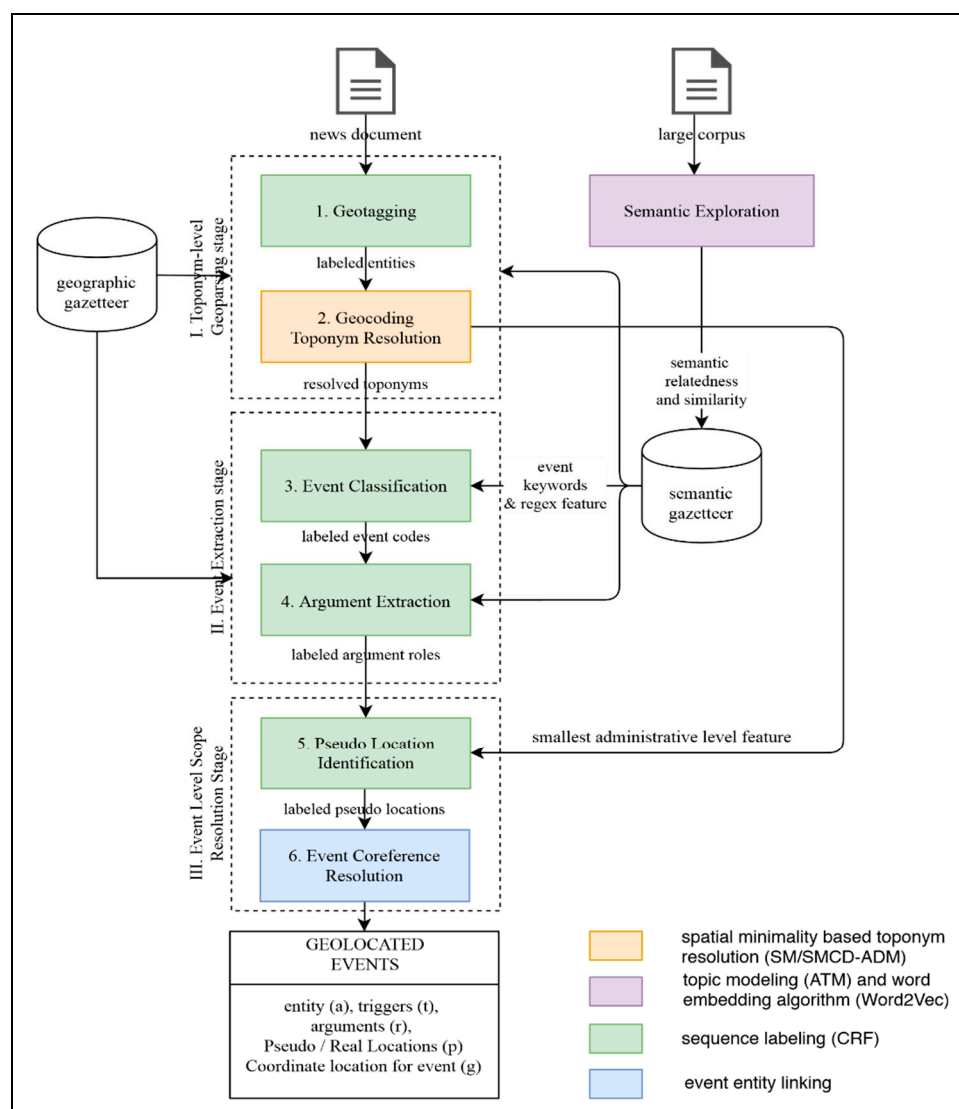


Figure 5. Integrated Event Extraction and Geoparsing: accept news document as input, resolving toponym and other entities (a), event triggers type (t), arguments (r), and event locations (g) from text. It is chaining geotagging with toponym resolution and event extraction. The system uses semantic gazetteer for features and regular expression rules learned from large corpus to increase the precision, recall, and geoparser accuracy.

Every token is then looked up and matched to a gazetteer entry which will provide gazetteer detection feature, so a positive match inside the gazetteer correlates positively with toponym detection although not necessarily deduced to a detected toponym (the inference will be done by the CRF inference layer). We are using Global Administrative Areas (GADM) database [69] for the main

reference for the gazetteer. The secondary gazetteer is the US cities list obtained from Simplemaps.com. It enlists US cities names under every state in US. The US cities data entries do not exist in GADM albeit it is very often mentioned in the text. The reason is that GADM in the US context only stops at the second level without having cities listed. For example, city of Prescott inside Arizona state does not show up in GADM database. The county where Prescott is located is Yavapai County, and it is present in the database. The typical pattern in the news, however, does not reference county name, so the augmentation of GADM is needed for US areas.

Similar to the approaches in many geotaggers, each sentence is then consulted to NLP Part-of-speech (POS) tagger, so there is an obtained POS tags for a better improvement of the tagging process.

For this purpose, we use InaNLP [70] that uses HMM based tagging for Indonesian language. The output of each word token within the sentence is a POS Tag derived from Penn's Treebank POS Tag standard. We then use LSTM-CRF as sequence labeler to perform the entity extraction (which simultaneously provide the functionality of geotagging) with the POS Tag and Gazetteer detection feature (as baseline features) added with (1) event keywords and (2) regular expression rule features that are obtained from semantic gazetteer which will be described shortly. In this setting, the fitting and the training is done sentence by sentence where every token in the input sentence (X) shall be mapped into the label token (Y).

The result from geotagging step is the following labels: LOCs (for each detected toponym) along with EVEs (event trigger), ARGs (event arguments can be numerical or string), and ORGs (named entity of organization). Each of these labels are prefixed with B and I, indicating beginning or inside the token, respectively. The output of this step is then carried forward to subsequent step to increase the later step performances.

The second step is the geocoding process. This is done by invoking an algorithm that is based on the toponym resolution algorithm Spatial Minimality [12]. Each of the LOC entities detected on the first step will be having a resolved geographic coordinate and also administrative level attached. From this process, we obtain a binary feature called Spatial Administrative Level. Both of these features and the toponym resolution algorithm are discussed in more detail in Section 4.5.

The event trigger classification step (step 3) is then commenced with entity features that have been extracted from an earlier step. The output (target variable) from event trigger classification is one of four major geospatial events tag for each EVE entities (ACCIDENT-EVENT, FIRE-EVENT, FLOOD-EVENT, and QUAKE-EVENT). This result will be subsequently fetched as an additional feature onto the Argument Extraction step (step 4) where each argument type (e.g., DeathVictim-Arg) is inferred for each ARG entity.

The next step (step 5) is Pseudo-location Detection, where every LOC entity is classified either as true location or pseudo-location one. The pseudo-location tags are also fit and tested using the results coming from earlier steps. However, as an important additional feature, we propose the use of *smallest administrative level* (SAL) feature to check whether a location entity is the smallest administrative level or not, in combination with other event semantics feature (event arguments and event types). This needs a result from the disambiguation (step 2) which uses geographic gazetteer and toponym resolution algorithm (SMCD-ADM). Note that all of these steps (with exception of step 2) involve the use of combination of neural and discriminative model LSTM-CRF architecture (coded as green boxes on Figure 5) and would require initial training first by fitting to the training set. The performance of the sequential labeling will be discussed in the Result section. The complete list of features used within these stages is listed in Table 4.

Table 4. Features for entity, event, argument, and pseudo-location identification.

Category	Feature Code	Type and Source	Features
Event Keywords	event	Semantic Relatedness from ATM	Binary feature Is_Event_Keyword(w): whether a word is included in shortlisted trigger words or not. Composed of four

(geospatial) events word bigrams list: flood, quake, fire, and accidents.			
Smallest Administrative Level (SAL)	sal	Geographical Feature	Is_SAL(t): whether a mentioned toponym has the smallest administrative level in the document
Gazetteer Detection	gaz	Geographical Resource: GADM database + US_Cities	Is_Toponym(w): a word is listed in Hierarchical Gazetteer or not.
Regex for detecting arguments	arg_regex	Semantic Similarity from Word2Vec & Semantic Relatedness from ATM	Regex Rules, composed of the following rules to detect patterns of these types: <ol style="list-style-type: none"> 1. Is_Time 2. Is_Plate_Number 3. Is_Coordinate 4. Is_Numeric 5. Is_Road 6. Is_Geographical 7. Is_Date 8. Is_Day 9. Is_Vehicle
Regex for detecting organizational entities	org_regex	Semantic Similarity from Word2Vec	Regex Rule, composed of the following rules to detect patterns of these types: <ol style="list-style-type: none"> 1. Place Types 2. Public Office Positions
POS Tag	postag	Syntactic Resource: INANLP	<ol style="list-style-type: none"> 1. First level POS Tag (e.g., NN) 2. Full level POS Tag (e.g., NNP) 3. Word Form: is_Upper 4. Word Form: is_Digit 5. Word Form: is_TitleCase
Entity	entity	Output labels from entity extraction step	LOC, EVE, ARG, ORG
Event	event	Output labels from event trigger classification step	FLOOD-EVENT, FIRE-EVENT, QUAKE-EVENT, ACCIDENT, EVENT
Argument	arg	Output labels from argument extraction step	(see Table 3)

4.3. Analysis of the Topic and Event Space: Tying Themes to Geospatial Referenced Text

With more than 44,000 unique document tags and counting almost 650,000 documents, our corpus offered a vast topic space [19], and we are mostly interested in the different types of geospatial events with their detailed attributes. As in every text document, there can be a lot of topics discussed in the news articles, each topic can have a typical characteristic: the semantics of information, the syntactic of delivering the information, the typical semantic roles of phrases within the sentences. These factors add up the dimensionality of the feature set. One of the popular ways to perform dimensionality reduction is the topic modeling model and its (mostly) unsupervised learning algorithms. LDA is the prominent and simple topic model which has grown into many derivations catering to different needs and characteristics. LDA is an unsupervised topic model and is commonly used to estimate topic distribution within corpus. However, since LDA is unsupervised and has no explicit tags, we base our work on LLDA, which is the supervised version of LDA with the document tags as the label.

In this section, we are proposing Aggregated Topic Model (ATM), a supervised learning approach from document tags that aggregates the partitions of (also supervised) Labeled LDA

(LLDA) [20] results into a single topic model. The labels from this supervised approach are taken from tags of each document in the corpus. The objective for ATM is to provide a topic modeling tool while also solving the memory requirement of LLDA when dealing with a very large number of tags, without sacrificing the coherence of the produced topic sets. LLDA posits a single topic-word distribution for each unique tag (label) that it found in the document, leading to a huge memory requirement for very large number (more than 10,000) of tags, in which case can be considered as an extreme multi-label classification problem [59].

This approach pushes the number of topics (K) to tens of thousands, given the traditional tool that typically only manage K within tens or in hundreds. Caution needs to be taken as having too many topics will typically result in over clustering topics into a small and highly similar clusters [61]; hence, one important element of ATM is the merging of topics which have the same labels.

Different topic labels having a similar top-words distribution can be found using *topic_sim* metric. This different topic label is still retained (not merged) and can serve as additional human-readable caption for each topic.

The ATM schema is described in notations that combine standard graphical model plate notation (Figure 6), extended with an aggregating process notion. We begin the description of ATM by some definitions, following the notation of [71]. Firstly, we define a set of topic models which is a collection of entire topic model partitions inferred by a labeled topic modeling training for N sessions where each of the sessions works on an equally sized partition of the dataset,

$$T = \{\Phi_1, \Phi_2, \dots, \Phi_N\} \quad (5)$$

Each topic set partition (Φ_i) itself is defined as a set of topics obtained from a partition of Labeled LDA training (dashed box on the Figure 6), each having K topic:

$$\Phi_i = \{\varphi_1, \varphi_2, \dots, \varphi_K\} \quad (6)$$

Each of the topic φ are further composed of term words which belong to that topic. In other words, a distribution of word probability given that topic,

$$\varphi_k = p(w|z = k) \quad (7)$$

Hence, each word has probability given we select a particular topic.

$$p(w|z = k) = \{P_{\varphi_k}(w_1), P_{\varphi_k}(w_1), \dots, P_{\varphi_k}(w_v)\} \quad (8)$$

We can implement φ as a dictionary; each of the entries is a unique word that has probability value. Next, we define the count of each topic and the document tag labels for each as follows:

$$C = c(\varphi_1), c(\varphi_2), \dots, c(\varphi_k) \quad (9)$$

$$\Lambda = \lambda(\varphi_1), \lambda(\varphi_2), \dots, \lambda(\varphi_k) \quad (10)$$

Note that $c(\varphi_k)$ is defined as count of words in any document (document m at word n) that has been assigned topic index k:

$$c(\varphi_k) = |\{z \mid z_{m,n} = k\}| \quad (11)$$

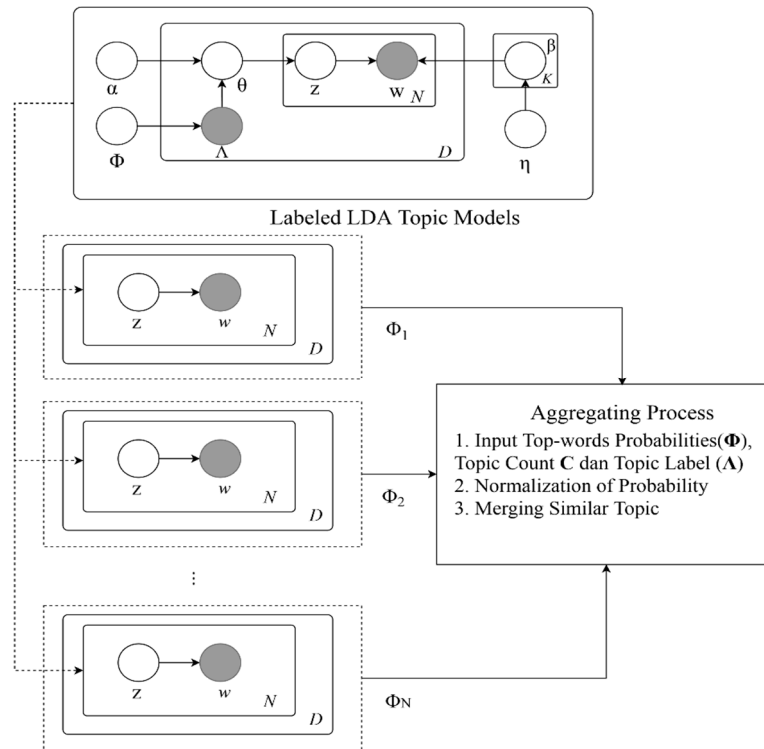


Figure 6. Aggregated topic model plate notation and schema.

Next, we are going to briefly describe the aggregation process to merge several labeled topic models into one. The aggregation process needs to use merging function between two topics that have the same labels (see Algorithm 1 (b)). The concept of merge is to recalculate the probability of each word component based on the weighted average of each word component given count of that topic (C). The output of ATM can be described as a semantic relatedness word vector, similar to the output of LDA/LLDA. However, ATM is able to manage all 44,280 unique labels in the main 650K corpus.

This merging function will be invoked from inside the aggregate function (see Algorithm 1 (a)), which essentially looks for any two or more topics which have the same label and merges them. The number of the assigned topic is represented by the area of the square (Figure 7). Each of the boxes is a topic (ϕ); the area is defined by $C(\phi)$ that is still decomposable by the (semantically related) keywords that are represented by the top-words w_1, w_2, \dots, w_v , variables which each have an area proportional to the probability of each word within that topic, $P_{\phi}(w_i)$. This provides a selection of words that, along with word embedding selection, comprise our event keywords and regular expression features.

<pre> procedure aggregate: input: T: set of topics $\{\Phi_{1..K}\}$ C: topic assignments count for all topic $\{c(\Phi_{1..K})\}$ A: set of labels of all topic $\{\lambda(\Phi_{1..K})\}$ output: merged topic model $M = \{(\varphi'_{1..K})\}$ begin: initialize M = {} for each topic $\varphi \in \Phi$: if label $\lambda(\varphi)$ exists in M: let $\varphi_{existing}$ where $\lambda(\varphi_{existing}) = \lambda(\varphi)$ $\varphi' = \text{merge}(\varphi, \varphi_{existing})$ append φ' into M else append φ into M, with adjusted $C(\varphi')$ end if end for end </pre> <p style="text-align: center;">(a)</p>	<pre> function merge(φ_1, φ_2): input: φ_1, φ_2: topics to be merged C: topic assignments count for all topic $\{c(\Phi_{1..K})\}$ output: new topic φ' begin: create new φ' which has all top-words from both φ_1, φ_2 let $C_{merge} = C(\varphi_1) + C(\varphi_2)$ for each $w \in \varphi_1$ and $w \in \varphi_2$: if w exists in both φ_1, φ_2: let $P'(w) = \frac{P_{\varphi_1}(w) \times C(\varphi_1) + P_{\varphi_2}(w) \times C(\varphi_2)}{C_{merge}}$ else if w exists only in φ_1: let $P'(w) = \frac{P_{\varphi_1}(w) \times C(\varphi_1)}{C_{merge}}$ else if w exists only in φ_2: let $P'(w) = \frac{P_{\varphi_2}(w) \times C(\varphi_2)}{C_{merge}}$ end if append w into φ' end for set $C(\varphi') = C_{merge}$ return φ' end </pre> <p style="text-align: center;">(b)</p>
---	---

Algorithm 1. Aggregate procedure (a) and Merge (b) function to form the aggregated model.

The aggregated topics will have all a unique set of labels (tags) from all documents. In order to see find the most similar topic that will be useful in exploring the semantic relatedness of the corpus, we adapt the standard cosine similarity for two vectors, making it appropriate in the context of topic models top-words vector. This similarity metric can be used to cluster similar topics and for taxonomy use is later demonstrated at Section 5.3.

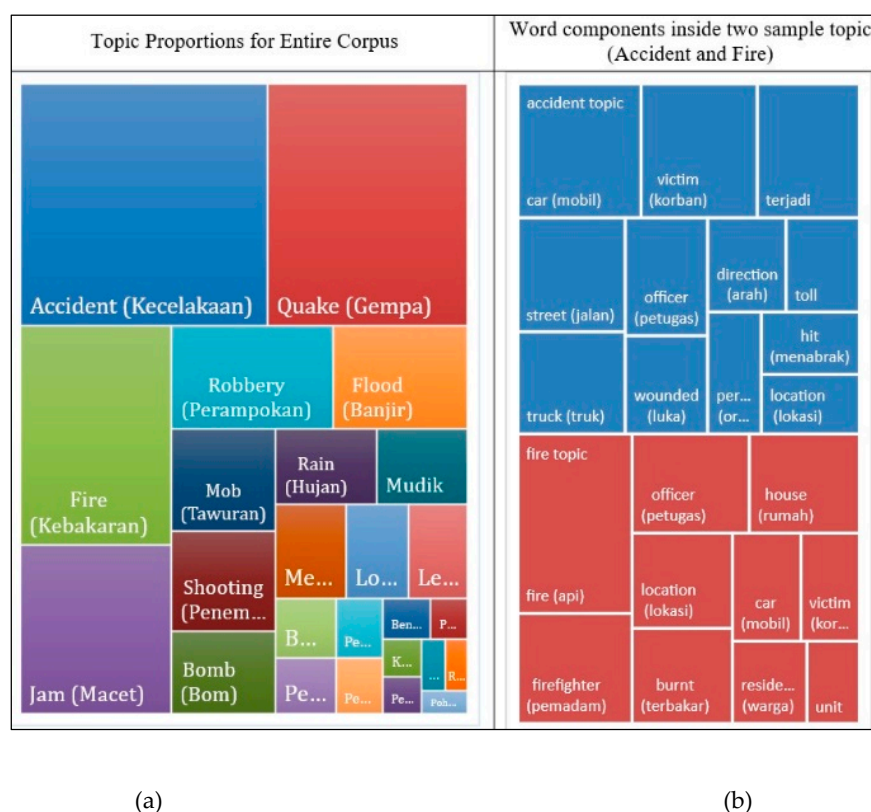


Figure 7. Treemap of Topic Proportions (a) and the top-words from Two Sample Topics (accident and fire) (b). The area shown on the left figure is determined by the number of topic assignments to that particular label/ $C(\varphi)$. The area shown on the right figure is determined by the probability of each word within that topic/ $P(\varphi|w_1)$.

4.4. Semantic Gazetteer for Event Keywords Feature and Numeric Argument Recognition

The large corpus provides wealth opportunity, for supervised or unsupervised learning, for mining semantic relations between words for adding generalizability of the model that was trained from the smaller, more detailed corpus [34]. We use Aggregated Topic Model to learn the semantic relatedness between topic label and words and *word2vec* word embedding to learn semantic similarity between words. The keyword extracts handpicked from these exploration models form the semantic gazetteer, which serves as a lookup method or list of terms with regards to various concepts (part of domain ontology). The term “gazetteer” here should not be confused with traditional geographic gazetteer that enlists place names. We used the gazetteer to build two derived features from it: (1) *event keywords* feature and (2) *regular expression* strings, which will be described as follows.

Event-keywords feature is a binary feature obtained from keyword lookup from a list of terms that is used as additional feature for generic classifiers designed for detecting event triggers and other arguments. For a matching keyword in the list, it will return “True”, otherwise it will simply return “False”. The structure of the Event-keywords feature is basically a set of lists of trigger keywords related to each major event that are obtained by selection of either top-words or most similar words or bigrams that have the most occurrences. The generated lists (see sample in Table 5) are created by three main methods, sorted by the probability or count, which will then be filtered manually:

1. Semantically related terms given a topic label, which is produced by our Aggregated Topic Model. (n-top-words).
2. Semantic similarity produced by Word2Vec [62] *most_similar()* function.
3. Bigrams counts produced by NLTK package n-gram analysis.

For example, the QUAKE-EVENT (“*gempa*” in Bahasa Indonesia) has the following set of keyword lists (Table 5). The generation of the words composing the list is automatic; however, it is filtered

manually for some words, that is, out of context or poorly generated. The calculated bigram is used mainly to supplement the I- (inside) entities detection. The first word in the bigram is the seed from the semantic relatedness and semantic similarity vector keywords (left and center column). The second word of the most counted bigram is then used as a feature for the labeling process.

Table 5. Event keyword-features for quake event.

Semantic Similarity Vector (Top 5)	Semantic Relatedness Vector (Top 5)	Bigrams and Count Vector (top 5)
semantic similarity of "quake"/gempa:	semantic related of "quake"/gempa:	bigrams of "quake"/gempa:
1. ("quake/gempa", 0.753)	1. ("shake/menggunca	1. ("earthquake/gempa
2. ("shake/guncangan", 0.739)	ng, 0.00177),	bumi", 3094),
3. ("tremble/getaran", 0.719)	2. ("repeated/susulan"	2. ("quake with
2. ("hurricane/topan", 0.710)	, 0.00029),	magnitude/gempa
3. ("richter", 0.693)	3. ("strength/kekuatan	berkekuatan", 1993),
	", 0.00013),	3. ("repeating
	4. ("scattered/berham	quake/gempa susulan",
	buran", 8.917e-06),	1062),
	5. ("cracked/retak",	4. ("volcanic
	8.916e-06)	earthquake/gempa
		vulkanik", 320),
		5. ("shallow quake/gempa
		dangkal", 27)

The semantic relatedness and similarity vector obtained from large corpus is also being used to build some regular-expression rule-based feature for entity and numerical argument recognition. This would improve the generalizability of the model, similar to the approach in [34]. An example of this feature is the `is_geographical(w)` argument feature as listed in Table 4, point 6. The function is basically a compiled regular expression pattern from the semantic gazetteer of geographical landmarks in Box 1.

Box 1. Example regular expression for recognizing types of place names. Terms separated by | (or) are composed from semantic similarity from names of rivers, settlements, and mountains, respectively.

```
(river|lake|sewer|riverbank|slope|ponds
|settlements|villages|area|farm|mount|mountain|caldera|crater)(\[A-Z\]\w+)
```

The next use of concept keywords within semantic gazetteer is to build a regular expression to recognize arguments from text. This will be the `arg_regex` feature that the sequence labeler will use. The inspiration is from RED/REDEX [72], although we do not employ learner model to learn regex from data. Instead, we are using the handcrafted regex similar to the output of that learner. The rule of the regex can be illustrated in the diagram below (Figure 8). The main component is the numerical expression stated via various regex string of "`\d`" character class followed by unit (e.g., *cm*, *meter*, etc.) The expression also accepts ranged expressions such as *(10-20 cm)*, of which the parser will take an average number later on. Moreover, a *string numeric* expression means that the regex will be able to detect patterns such as *"tens of victims"*. The capture group can be started or ended with role string such as *"the height of"* or *"person killed"*, which will translated to Height-Arg or DeathVictim-Arg by the argument extraction step. Some vague unit expression is also added to model notion of estimates such as *"knee deep"*. Note that instead of using regex directly to extract the values, we are

using regex to build a feature to detect which portion of the document matches the argument for a particular event. The feature will be used by the sequence labeling framework. The reason is that the statistical sequence labeler will do more generalization and less “brittle” inference.

4.5. Smallest Administrative Level (SAL) Geospatial Feature for Pseudo-Location Identification

To address the problem of discriminating true location entities (LOC) to pseudo-location (PLOC) entities, we develop a feature which exploits results from the toponym resolution process, i.e., the smallest administrative level. The motivation assumes that news article will report the most precise toponym possible to report the location of the event. We first obtain the administrative level from all disambiguated place names. Then, we can find the maximum level for a document level. The motivation behind this feature is to prefer a precise location more than an imprecise location; hence, a level 2 administrative such as city names (Bandung, Jakarta) is more precise than the provincial level (level 1). However, this feature will be combined with the event semantic labels (i.e., event type labels and event argument labels) from the earlier stages so that the classifier algorithm can make prediction based on the peculiarity for particular event types. The consideration is that we observe events such as Earthquake, which tend to occur or affect several provinces or even countries; hence, larger administrative toponyms mentioned in the text can be seen as true location entities instead of PLOC. The feature is referenced as Smallest Administrative Level (SAL) within document scope that is resolved by the disambiguation process for each toponyms found in the document using spatial minimality (SM) (see Algorithm 2) and spatial minimality centroid distance administrative (SMCD-ADM, Algorithm 3). SMCD-ADM is our modification derived from the elegant Leidner’s Spatial Minimality framework where:

(1) The area calculation is replaced by the calculation of distance of points to its centroid (*Centroid Distance*). This is useful for speeding up the process and to avoid the degenerate cases where there are only two or less toponyms inside the document. In other words, the minimality of area is replaced by the minimality of the distance of polygon candidates to its centroid (see Figure 9).

(2) The minimality of distance is adjusted by multiplying it by the administrative level of an area. Hence, the smaller administrative is a candidate referent, the less preferred it is. Note that this is the reverse principle from the smallest administrative feature to find out the smallest administrative area. This is because in this toponym resolution task, what is sought is the commonality of toponym mention, instead of the precision of the place mention on the Pseudo-location Identification task.

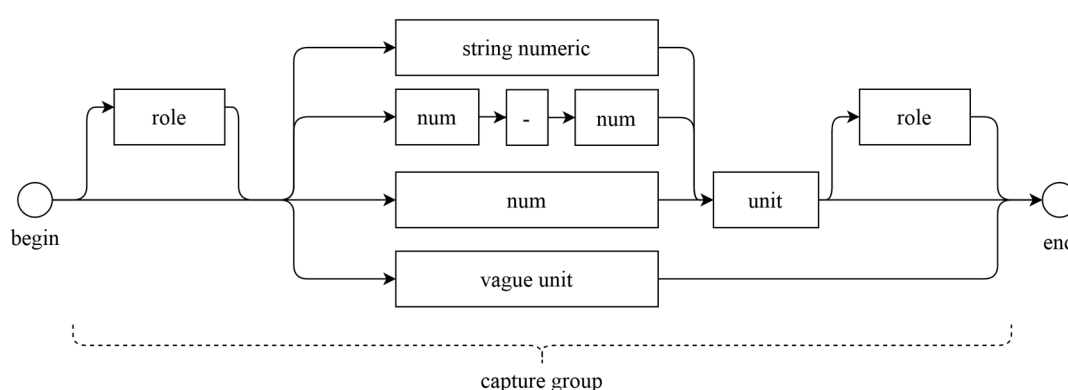


Figure 8. Regular expression to detect numerical argument. The argument typically either started or ended with the role keyword followed by various numerical quantities, followed by the unit of the argument. For example, “the accident left 2 people killed” will be extracted as 2 (numeric) people (unit) killed (role).



(a) **Spatial Minimality (SM)** on degenerate case. In the map, the toponyms that need to be disambiguated are “Banten” (A), “Jakarta” (B), and “Serang”, which have the two referents of “Serang, Banten” (C₁) and “Serang, Pekalongan” (C₂). The Area constructed by ABC₂ is much smaller than ABC₁ even though C₂ is much farther due to the degenerate polygon of ABC₂ which looks like a single line.

(b) **Spatial Minimality Centroid Distance (SMCD)**. Instead of preferring referent tuple, which constructs minimum area, SMCD prefers the minimum distance to the centroid for each tuple. In this strategy, there is no calculation of area as in original SM, so degenerate polygon cases can be avoided. In the above case, the ABC₁ was correctly selected because $d_1 < d_2$.

Figure 9. Illustration of difference of strategies between original Spatial Minimality (a) and Spatial Minimality Centroid Distance (SMCD) on the (b). The SMCD-ADM is SMCD but with adjustment on the weight factor of the distance.

<p>function getSmallestAdministrativeLevel (D: document, G: gazetteer):</p> <p>output: smallest administrative level of the document</p> <p>begin:</p> <p> initialize toponyms T = {}</p> <p> T = extract location entities from D</p> <p> DT = DisambiguateDocumentSM (T, G)</p> <p> L = {}</p> <p> for each t in DT :</p> <p> adm_level = lookup administrative level of t from G</p> <p> append adm_level to L</p> <p> return maximum adm_level from L</p> <p>end</p>	<p>function DisambiguateDocumentSM(T: list of toponyms, G: gazetteer):</p> <p>begin</p> <p> for each t in T:</p> <p> let $\tau(t)$ = lookup set all possible candidate-referents tuples from t in gazetteer</p> <p> let S = cross product of $\tau_1 \times \tau_2 \times \dots \times \tau_n$</p> <p> for each N-tuple C \in S do:</p> <p> H = polygon from all centroids in C</p> <p> A = Calculate area of H</p> <p> return tuple C* that has minimum A from all tuple C</p> <p>end</p>
---	--

Algorithm 2. Algorithm for finding Smallest Administrative Level feature using Spatial Minimality (SM) from [12].

Note that the smallest administrative level corresponds to the maximum integer indicated on administrative level field in the case of our chosen gazetteer (GADM) (the bigger the code number, the smaller region. Currently the largest number is 4, indicating village administrative level). Then, the binary feature is calculated by simply comparing whether the particular token toponym’s administrative level equals the smallest administrative level or not. The feature makes use of the output of spatial minimality algorithm to disambiguate document from the detected toponyms. Hence, basically it uses geometric minimality heuristics.

```

function DisambiguateDocumentSMCD-ADM(T: list of toponyms, G: gazetteer):
begin
  for each t in T:
    let  $\tau(t)$  = lookup set all possible candidate references from t in gazetteer G
  let S = cross product of  $\tau_1 \times \tau_2 \times \dots \times \tau_n$ 
  for each N-tuple C  $\in$  S:
    Cd = calculate centroid of all points in C using G
    maxP = find point p  $\in$  C that has maximum distance to centroid Cd
    maxdistc = distance of maxP to centroid Cd
    adm_levelc = administrative level of maxP
    adjusted_maxdistc = (adm_levelc + 1)  $\cdot$  M  $\cdot$  maxdistc
  return tuple C that has smallest adjusted_maxdistc
end

```

Algorithm 3. Modified Spatial Minimality with Centroid Distance and adjustment factor based on Administrative level and adjustment constant M (SMCD-ADM).

5. Experiments and Results

As indicated earlier we approach the geotagging and event extraction as a sequence labeling problem. Geotagging problem in this work is cast as a subset of entity extraction, extracting the LOC entities as toponyms for the further steps. The entity extraction, event classification, argument extraction, and pseudo-location detection steps make use of the Conditional Random Field sequence labeler from the NCRF++ toolkit [73]. We configured a CRF inference layer that sits on top of (bidirectional) LSTM word sequence layer and did not use any character sequence layer. The LSTM layer functions as feature extractor, while the CRF is set up to capture dependencies of neighboring labels. We chose the BiLSTM-CRF as it is currently one of the state-of-the-art model combinations [74], replacing regular linear chain CRF in our earlier attempt. We also used Glove [32] word embedding vector trained from the corpus on the bottom layer. Adam optimization (included in NCRF++) is used for all of the training session. Most importantly, all combination of features listed on Table 4 tested and fetched as handcrafted features to the NCRF++ training setting. The training of this model then commenced with 927 sentences, 16,444 tokens on subset of the large corpus with four main topic categories: Quake (24%), Accident (21%), Flood (30%), and Fire (25%). We evaluated the standard definition of precision, recall, and the F1-score on each of the steps above.

5.1. Geotagging

For the entity extraction, we then compare the model with baseline LSTM-CRF with gazetteer and POS tag features without including the event-keyword features and regular expression argument extractor. The inclusion of the two features is seen as a reasonable improvement. A similar approach is also taken for Pseudo-location detection. For the detailed set of features, please refer to Table 4. The entity extraction result is summarized in Table 6.

Table 6. Entity extraction performance (step 1).

Entity	(Baseline) LSTM-CRF with Gaz + Postag Features			(Proposed) LSTM-CRF with Org_regex + Arg_regex + Ev_Keywords + Gaz + Postag Features		
	P	R	F1	P	R	F1
LOC	0.929	0.897	0.912	0.951	0.897	0.923
ARG	0.762	0.767	0.764	0.857	0.709	0.776
ORG	0.697	0.847	0.765	0.787	0.847	0.816
EVE	0.850	0.888	0.869	0.855	0.925	0.889
micro avg	0.797	0.830	0.813	0.863	0.811	0.836

<i>macro avg</i>	0.809	0.850	0.828	0.863	0.845	0.851
<i>weighted avg</i>	0.801	0.830	0.814	0.865	0.811	0.834

Event extraction stage result which is composed from event trigger classification step (Table 7) and event argument extraction step (Table 8) is done by training CRF again, but with the predicted Entities fetched from the earlier Entity Extraction step.

Table 7. Event trigger classification performance (step 3).

Event	(Baseline) LSTM-CRF with Gaz + Postag Features			(Proposed) LSTM-CRF with Entity Features		
	P	R	F1	P	R	F1
ACCIDENT-EVENT	1.000	1.000	1.000	0.962	1.000	0.980
FIRE-EVENT	0.806	0.967	0.879	0.968	1.000	0.984
FLOOD-EVENT	0.886	0.861	0.873	1.000	0.972	0.986
QUAKE-EVENT	0.885	0.793	0.836	1.000	1.000	1.000
<i>micro avg</i>	0.885	0.900	0.893	0.983	0.992	0.988
<i>macro avg</i>	0.894	0.905	0.897	0.982	0.993	0.987
<i>weighted avg</i>	0.889	0.900	0.892	0.984	0.992	0.988

Table 8. Argument extraction performance (step 4).

Event	(Baseline) CRF with Gaz + Postag Features			(Proposed) CRF with Entity + Event Features		
	P	R	F1	P	R	F1
DeathVictim-Arg	0.615	0.381	0.471	0.760	0.905	0.826
Vehicle-Arg	0.625	0.435	0.513	1.000	0.913	0.955
Height-Arg	0.875	0.700	0.778	0.833	1.000	0.909
OfficerOfficial-Arg	0.711	0.678	0.694	0.849	0.839	0.844
Time-Arg	0.927	0.962	0.944	1.000	1.000	1.000
Place-Arg	0.873	0.832	0.852	0.900	0.884	0.892
Street-Arg	0.708	0.810	0.756	0.526	0.952	0.678
Strength-Arg	0.952	1.000	0.976	1.000	1.000	1.000
<i>micro avg</i>	0.822	0.766	0.793	0.869	0.912	0.890
<i>macro avg</i>	0.786	0.725	0.748	0.859	0.937	0.888
<i>weighted avg</i>	0.812	0.766	0.785	0.884	0.912	0.894

The above results displayed the baseline performance vs. highest performance of particular combination of features for each step of the event geoparsing which use sequence labeling (steps 1, 3, 4) with the exception of step 5. We are separating the result of step 5 due to its central importance in this process. To see which features combinations contribute the most to the performance of the system, we conducted the *ablation test* for each of the four sequence labeling steps where sequence labeling is applied. There are 9 features in total to be tested, of which some subset of possible feature combinations feature the label displayed on the leftmost column (testing and analyzing all 2⁹ combinations is prohibitive for our resource). The enabled features are represented by blue box, while the disabled features are represented by grey box. The performance of the particular combination is displayed in the chart with the range of weighted F1 score performance (based on enabled features) between 0.65 and 0.9 (vertical axis on the top graphic of Figure 10). The entity label produced by the entity extraction step is referred as *entity* feature. Similarly, the result for the event trigger classification step is called *event* feature, and the result from argument classification is *argument*

feature. The *arg_regex* and *org_regex* is both the regular expression feature derived from keywords from semantic gazetteer, for the detection of numerical argument and organization, respectively.

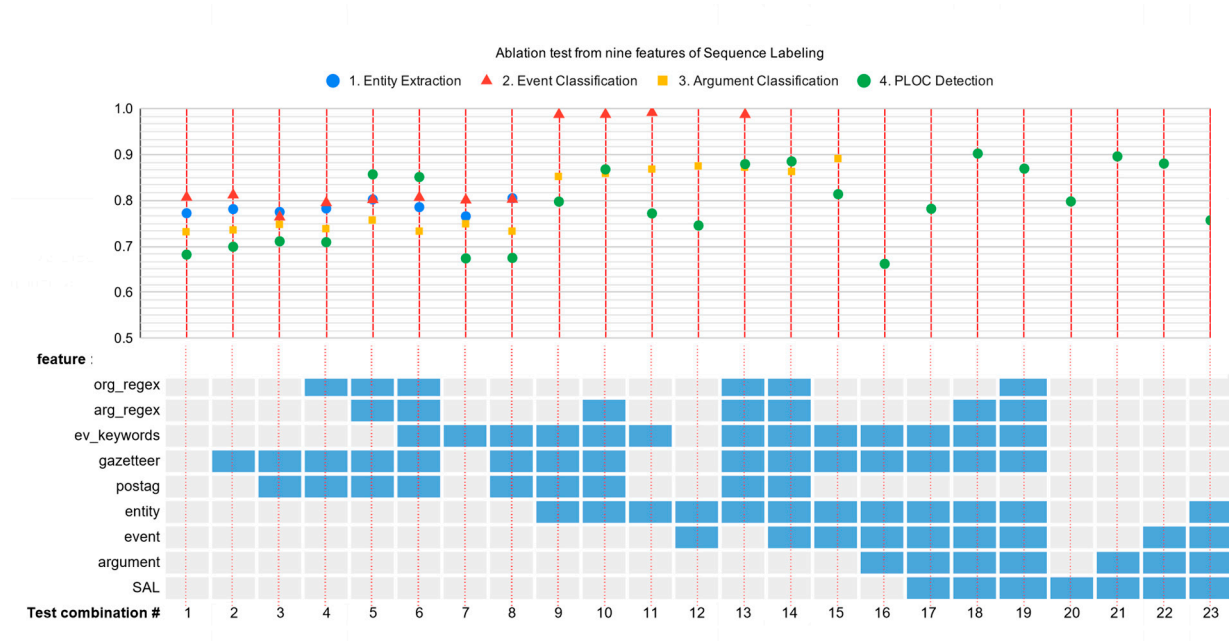


Figure 10. Ablation Test of weighted F1 score from nine combinations of features of four geotagging steps (step 1, 3, 4, and 5 from Figure 4). Active features are marked as blue cells (below part of the graphic). Missing score points means such combination of features is not applicable on that particular step.

5.2. The Pseudo-Location Classification

In this fifth step experiment setting, the objective is that every toponym in the corpus is attached a correct label, indicating whether it is a valid, precise toponym that serves as true locational reference label (LOC) or a pseudo-location (PLOC). This is the $p_i^{(k)}$ variable explained in Section 4.1. From the ablation test, the use of geospatial information of SAL feature (Section 4) is very effective to boost the F1 score. The combination of *argument* and *event* feature with SAL feature will add to the performance by a significant margin. This shows that event semantics can actually aim to the identification of pseudo-location entities, which is a crucial task in our event geoparser model. The result of this step is presented on Table 9.

Table 9. Pseudo-location identification (step 5).

Tag	(baseline) LSTM-CRF with Gaz + Postag Feature			(Proposed) LSTM-CRF with Sal + Event + Arg Feature		
	P	R	F1	P	R	F1
PLOC	0.835	0.784	0.809	0.971	0.879	0.923
LOC	0.528	0.655	0.585	0.809	0.948	0.873
micro avg	0.713	0.741	0.727	0.908	0.902	0.905
macro avg	0.681	0.720	0.697	0.890	0.914	0.898
weighted avg	0.733	0.741	0.734	0.917	0.902	0.906

To provide a more illustrative case for this task, we inspected the output from step 3, step 4, and step 5 of the event geoparsing workflow and found interesting instances of pseudo-location identification. One of the labeled sentences is displayed on Figure 11. The sentence is a news snippet about an accident of a trailer truck in Demak regency which happened while on its way to Kudus regency. Geotagging step and geocoding step of the first stage and second stage have been performed, and we are focusing to the Step 5 (of the third stage) of this discussion. On this step, if we

remove the information from the event semantics (event argument and event trigger features), the geoparser fail to see that Kudus is not the location of the accident event (i.e., non-locative). Both are seen as valid literal toponyms. Thus, it labeled both toponyms as correct locations of the accident event (LOCs) whereas Demak is the *locative* one. However, with the inclusion of a feature from Step 4 (arg) and Step 3 (event), the geoparser correctly identified the real location of the event). In particular, step 4 produced the label for Kudus FromTo-Arg (the origin or destination of the vehicle) instead of Place-Arg, indicating destination instead of location of event. This is in accordance to our observation in the news stories that the event semantics for accident (ACCIDENT-EVENT) often has such argument role (the supposed destination of vehicle). The Central Java (Jawa Tengah) is the province of both cities, and it is also a correct, locative toponym. However, due to lack of precision of that toponym (i.e., not precise), it is correctly marked as PLOC. To be able to differentiate this, the inference algorithm was assisted by information about the smallest administrative level of Step 2, which requires the use of toponym resolution algorithm with hierarchical gazetteer due to some ambiguities of Kudus and Demak. With the proposed workflow, the final stage of the process then resolves the accident event location coordinate to City of Demak, Jawa Tengah (−6.875, 110.652) by identifying out two PLOCs (Kudus and Jawa Tengah).

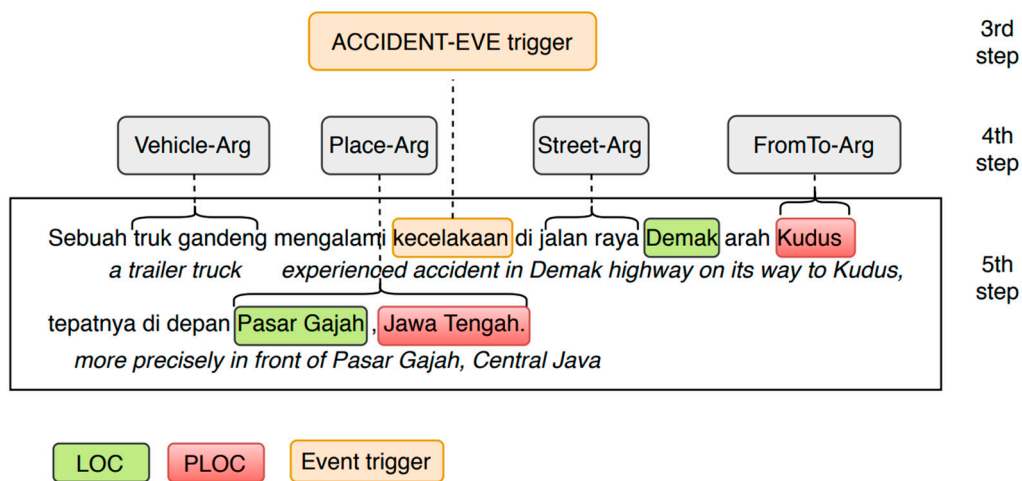


Figure 11. Event geoparser correctly assigned the real LOC label to Demak and pseudo-location PLOC label to Kudus with the help of event argument (step 4) and event trigger feature (step 3) outputs. Entity tags from Step 1 output are omitted for clarity.

5.3. Aggregated Topic Model

There are two main variants of LDA solver that we use, the Gibbs Sampler and Variational method. MALLET implements Gibbs sampler while the Gensim toolkit uses Variational method. Gibbs sampling generally provides better quality of topic model. The quality of topic model can be measured using some different metric. The earliest method uses perplexity metric [58] while the latter works often use the topic coherence metric, introduced in [75]. The one used in this experiment, topic coherence, is a metric that measures the quality of the produced topic model given by the co-occurrence of the top words in a particular topic. The more coherence scores towards zero, the higher the probability of co-occurring top-words of a topic within the corpus; thus, it generally means the higher quality of the topic discovered. The topic coherence metric (UMass) is described as

$$Coh(t, V^t) = \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{D(v_m^t, v_l^t) + 1}{D(v_l^t)} \quad (12)$$

where $D(v)$ represents document frequency, i.e., the number of documents that has word v at least once. $D(v_1, v_2)$ is the co-document frequency, defined as a number of documents which have both words v_1, v_2 . Thus, the coherence metric (Coh) is calculated based on co-document frequency of each m top words pairs for topic t .

We compared the coherence metric using the following approach:

1. LDA implementation of MALLET (LDA via Gibbs Sampler) [76];
2. LDA implementation of Gensim (LDA Variational Bayes) [77];
3. Labeled LDA (LLDA) code that is implemented inside MALLET [76];
4. Aggregated Topic Model.

The result from this comparison is listed in the table above (Table 10). In terms of coherence metric, our proposed method is placed better than LDA VB K = 600 and LDA Gibbs K = 100. However, K is much higher than the counterpart. The Labeled LDA that was tested on our system (32 GB RAM) crashed due to insufficient memory if being initialized with K more than 15,000 labels.

Table 10. Topic coherence metric from topic models (lower coherence score is better).

Top words	Model	K	Coherence	
			Flood Topic	Quake Topic
20	Labeled LDA (LLDA) (15K only)	2,588	−201.01	−187.46
20	Aggregated Topic Model (ATM)	44,280	−393.96	−394.31
20	LDA Gibbs K = 100	100	−421.87	−424.72
20	LDA Gibbs K = 600	600	−285.51	−397.41
20	LDA VB K = 600	600	−453.82	−417.77

To explore some thematic space from the corpus, we are interested in obtaining some taxonomy for popular topics. The resulting proposed topic model from the corpus can easily be queried for the top words based on the topic label obtained from document tag (ϕ_k) and also the topic similarity using the algorithm. From the seed topic label (for example “jakarta flood”/“banjir jakarta”), we limit to the five most similar topics, each having ten of their top-words as a cut off. The result is then displayed as a tree structure in Figure 12. Obtaining this result is not doable straightforwardly from the Labeled LDA because the memory limitation on the number of unique document tags.

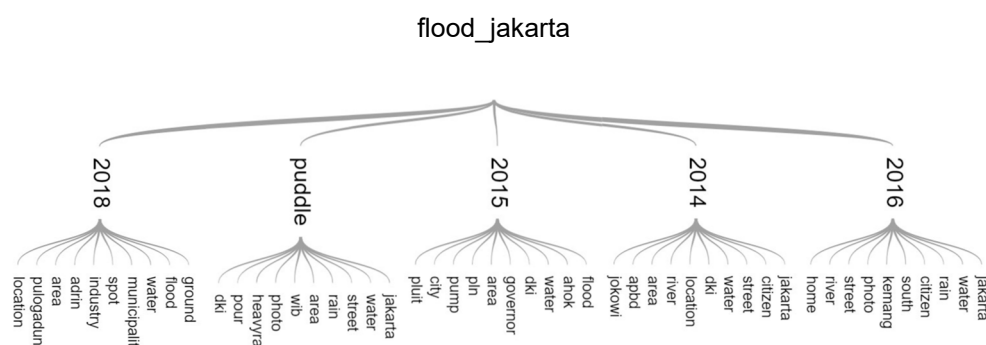


Figure 12. Taxonomy generated (translated) by topic similarity metric from seed root node (topic tag) “banjir_jakarta” (jakarta_flood). The leaf nodes are the top words of their respective parent topic. The generated tree is limited to the 5 most similar topics.

5.4. Disambiguation and Toponym Resolution

We test the SMCD-ADM with the baseline disambiguation method based on *spatial minimality heuristic* introduced by Leidner [12]. We also use the *one-referent-per-discourse heuristic*, meaning that several instances or tokens of the same toponym will be resolve to a single referent throughout document. The accuracy is calculated by dividing the correct disambiguation with the number of toponyms tested. The number of unique toponyms tested is slightly different, as there is a limitation that spatial minimality cannot work with less than three points in the candidate tuple. The result is presented on Table 11.

Table 11. Toponym resolution performance (accuracy) (step 2).

Algorithm	Spatial Minimality	SMCD-ADM
Toponyms tested	791	792
Correct Disambiguation	561	588
Accuracy	0.70	0.74

5.5. Auto Generation of Rich Thematic Map from Single Article

The last experiment is more of an exploratory task which captures the information in form of thematic choropleth map. The task is to fetch text of the flood topic through the entire (extended) event extraction geoparsing workflow, obtaining tagged entities, event triggers, arguments, and pseudo-locations. The article of our choice contains several events of the same type at several places, and each has numerical arguments describing measurements of the event. From our observations, these types of articles are pretty common in the corpus.

The numerical arguments and the location entities (after discarding all pseudo-locations) are linked through the same sentence index, and the arguments are extracted and parsed and appropriately projected onto the map only from a single document. In the case of the article about flood report, the main arguments are the height of the flood (Height-Arg, in centimeters) in several areas in Jakarta. If there are several numbers within the span of the argument, these numbers will be averaged before they linked to a particular location. We use Geopandas toolkit for visualization of the thematic map using the extraction result and filter the query with geo dataframes in South Jakarta and East Jakarta.

The basemap was provided from GADM all countries data. The overlay waterway data of river Ciliwung (blue line) is obtained from *petajakarta.org*. The extraction visualization result can be seen in the diagram on Figure 12.

6. Discussion

In the first and second experiment, we are testing the combination of features for the three stages of event geoparsing. The first stage can be considered as standard geotagging using NER with some help from POS tagger component. In the first step, even though the model is equipped with event keywords features and regular expression rules compiled from semantic gazetteer, it had improved entity recognition by a small margin of 2.46% (weighted F1 on first entity extraction step on Table 6). The second step (event trigger classification) resulted in 10.76% improvement (see Table 7). In the third step (argument extraction on Table 8), the improvement margin was 13.88%. The small improvement margin in the first step can be explained to the relatively standard entity extraction task, which can already be performed well with existing methods. However, as we continue along the downstream stages (which use features from the earlier stages including semantic labels such as event label and event arguments), the results gained get more significant. Thus any accuracy gained in the earlier stage is important to the downstream stage, as observed in many extraction works (e.g., [34] or [78]). This is much more apparent to the last stage which is arguably the centerpiece of the event geoparser requirement to separate pseudo-location from the real location of the event. It can be seen that the pseudo-location identification task had been improved significantly in order to discriminate the true location of an event vs. its pseudo-location. The use of event semantics, i.e., event labels (from Step 3) and argument (from step 4) combined with geospatial feature (SAL from Step 2) eventually improved the performance by a substantial margin. From the ablation test we tested, if we use only dd SAL geospatial feature, it will only increase around 6.2% from the baseline performance. If we include the argument feature, it would add more significant performance, up to 22.9%. Including the event feature will further increase the F1 score, outperforming the baseline gazetteer and postag feature by 23.43% margin. This shows that event semantics supplied by the event extraction methods from our proposed event geoparsing stages are able to improve geoparsing with event-level scope resolution. The inspection of the event geoparser's output from Section 5.2

also supported this hypothesis, which also answers the problem posed in Section 2.3 regarding the ability to identify both locative and precise LOC entities.

The list of event keywords and both of the arg/org regex features that had been derived from semantic gazetteer are able to improve the recall as they provided related and similar keywords that might not be seen in the development set, thus preventing overfitting that might hinder model generalization. The inclusion of those features only works well for the first and second stages of the extraction (approximately 2.77% and 4.44%, respectively).

The third experiment shows that Aggregated Topic Model (ATM) can serve as an alternative topic model due to the capability of holding a large number of K within the Labeled LDA setting. This is especially useful when dealing with memory problem of LLDA with a large number of labels (extreme labeling problem) that we often find in web news portals or social media. The ATM can still provide decent coherence, even better than LDA (Gibbs sampling version with $K = 600$), despite the large number of topics that it needs to handle. The coherence of ATM, however, is less than LDA or LLDA with lower K setting. From the ablation that we conduct in Section 5.1, the addition of the handcrafted feature that uses information from keywords derived from semantic exploration added performance around 3–19% for each step.

The event extraction framework used in this work is still using local, per sentence features, except (1) the tags result for each step and (2) the SAL of the document where the feature must be computed per document (global) after a toponym disambiguation is performed. The work of [78] and [34] uses global features and a joint model to perform the event extraction task, and its integration is worth to pursue. Moreover, it is worth to mention that the task of event extraction can be structured (due to its similarity) as dependency parsing task, with semantic roles representing the dependent entities to the event anchors or trigger [79].

With all the event geoparser components put in place, we then have the choropleth map visualized automatically for flood topic (Figure 13) on a single document. Darker tone means higher water level (Height-Arg), which is only one of the argument types extracted (along with number of AffectedVillage-Arg and other numerical arguments). The Cause-Arg is also extracted with value “Kali Ciliwung” (Ciliwung River), which is represented by the blue line overlaid on top of the choropleth map. The interplay between extracted event semantics and inferred geospatial location provided can be seen. This may be used as richer data for generating various thematic maps and further geospatial analysis. Arguably, looking at the thematic map is easier and faster for delivering geospatial information across to the human reader. The map can be considered as an exploratory analysis to augment the geospatial event information presented in text and gives the reader a better understanding of it. The ability to efficiently extract and map information from a single document without the need of multi-documents aggregation or retrieval methods shows a potential use case of event-level scope resolution geoparser.

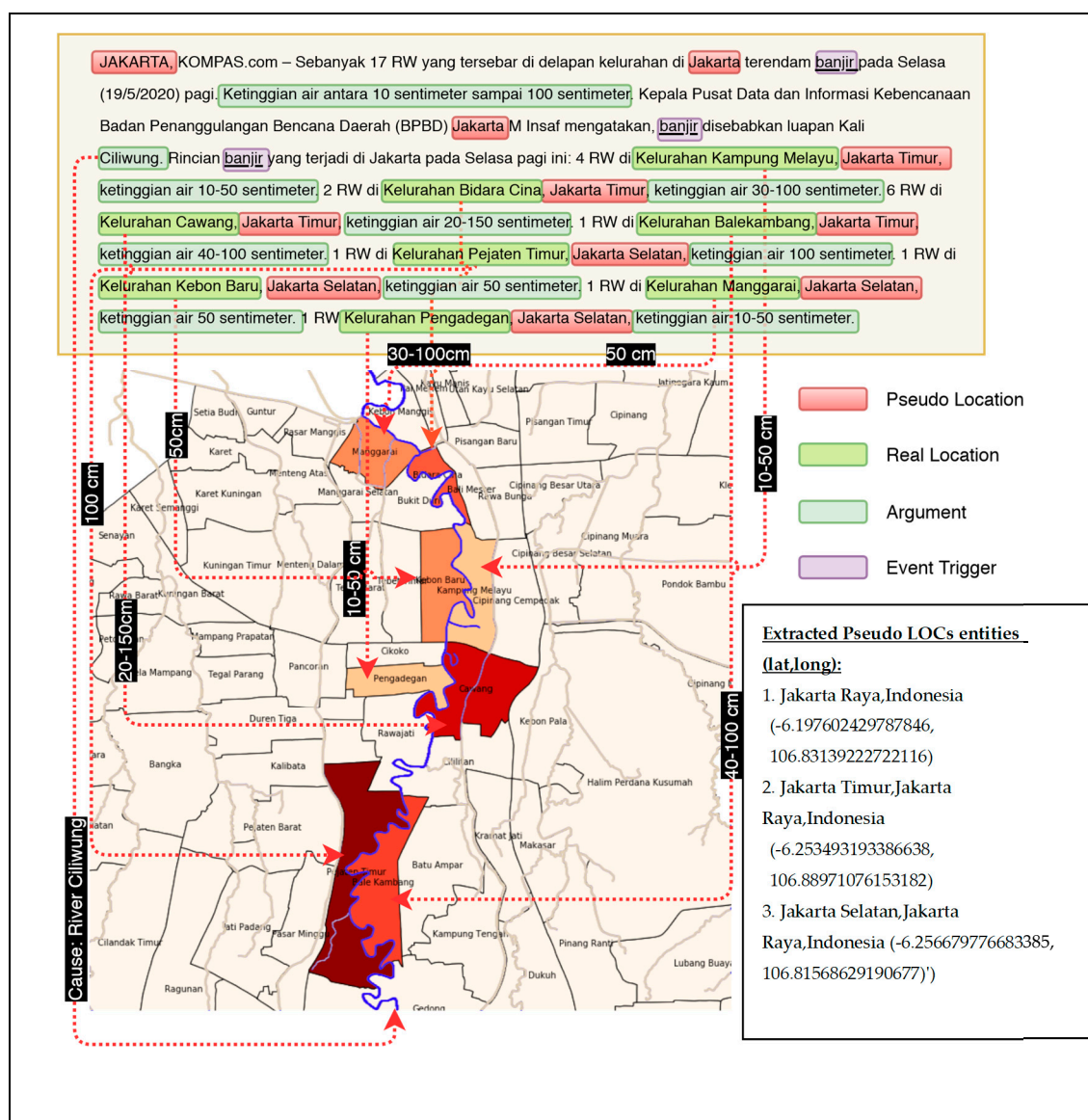


Figure 13. Visualization sample from a single article with result from our proposed Event Geoparser. The source article is displayed and tagged with colors, indicating arguments, event trigger, and pseudo-location/real-location label for every detected toponym.

7. Conclusion

Geoparsing and event extraction are both active research topics and have been around for more than a decade. The recent works on geoparsers are more equipped with natural language processing and machine learning techniques to better cope with the sheer size of unstructured text data. However, even in the modern geoparsers landscape, little has been studied on integration of geoparsing with event extraction framework (or vice versa) for the event geolocation needs, especially in dealing with the resolution on the event-level scope where existing geoparsers are only coupled with independent event coder component in a separate, opaque fashion.

The work described in this paper described a novel approach that tightly integrates geoparsing and event extraction in three stages. In particular, it shows how the integration of event semantics with geospatial based features benefited the event geoparsing workflow by substantially improving the pseudo-location identification which is crucial to the task of resolving the event-level resolution scope. The integrated event extraction framework provides event semantics (event types and arguments) which is beneficial to the main goal of event geolocation, which also enables the extraction of numerical arguments at particular disambiguated toponym, which provides richer semantic context for further processing. This in turn would be useful for many Geographical

Information Retrieval applications, as suggested by the thematic map generation example only from a single document.

We also augmented the geoparser with the Aggregated Topic Model as a semantic exploratory tool from a large multilabeled corpus, which is typical on the news sites. The ablation test shows that the event keywords derived from ATM and word2vec are able to improve the generalizability of the model. The coherence test shows an acceptable performance of ATM even with a very large number of topics (K). Thus, it is a valuable tool for exploring semantic relatedness especially with multi labeled corpora.

Moreover, contributed by this work is the event geoparsing news corpus in Bahasa Indonesia, which offered a new testbed for extraction of events and event arguments along with geoparsing task. This may serve to expedite the research of further event extraction framework. Even though the domain for this geoparser is news articles in Bahasa Indonesia, we believe that the proposed event geoparsing model is useful in other languages as well, given a good enough corpora. In the future, we plan to develop a pipeline which integrates a visual GIR system as additional component to the extraction and geoparsing method described here, and which serves automatically generated thematic maps from attribute data. In terms of the model's architecture, casting (most of) event geoparsing tasks as pipeline of sequence labeling tasks which will be solved by LSTM-CRF model works well with the categorical corpus. It may be supplanted by joint, structured prediction models for better performance. This particular result served as evidence that integration of geoparser method (disambiguation to the correct administrative level and coordinate) with event extraction technique is useful to resolve geoparsing at event-level scope of resolution. Lastly, it must be noted that the integrated event extraction as described will add several layers of processing. This may be a disadvantage in terms of runtime of execution of the model, especially in large scale settings such as in GDELT or ICEWS scale.

Author Contributions: Agung Dewandaru is responsible for the conceptualization of the research, implementation and experiments. Dwi Hendratmo Widyantoro envisaged the extraction process and advised about the revised neural method. Saiful Akbar contributed on the result analysis and error analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by P3MI-ITB program.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Himmelstein, M. Local search: The Internet is the Yellow Pages. *Computer* **2005**, *38*, 26–34, doi:10.1109/MC.2005.65.
2. Wunderwald, M. NewsX: Event Extraction from News Articles. Master's Thesis, Dresden University of Technology, Dresden, Germany, 2011.
3. Gelernter, J.; Balaji, S. An algorithm for local geoparsing of microtext. *GeoInformatica* **2013**, *17*, 635–667, doi:10.1007/s10707-012-0173-8.
4. Wang, W.; Stewart, K. Spatiotemporal and semantic information extraction from Web news reports about natural hazards. *Comput. Environ. Urban. Syst.* **2015**, *50*, 30–40, doi:10.1016/j.compenvurbsys.2014.11.001.
5. Freifeld, C.C.; Mandl, K.D.; Reis, B.Y.; Brownstein, J.S. HealthMap: Global Infectious Disease Monitoring through. *J. Am. Med. Inform. Assoc.* **2008**, *15*, 150–157, doi:10.1197/jamia.M2544.Introduction.
6. Purves, R.; Clough, P.; Jones, C.B.; Arampatzis, A.; Bucher, B.; Finch, D.; Fu, G.; Joho, H.; Syed, A.K.; Vaid, S.; et al. The design and implementation of SPIRIT: A spatially aware search engine for information retrieval on the Internet. *Int. J. Geogr. Inf. Sci.* **2007**, *21*, 717–745, doi:10.1080/13658810601169840.
7. Gritta, M.; Pilehvar, M.T.; Collier, N. A pragmatic guide to geoparsing evaluation. *Lang. Resour. Eval.* **2020**, *54*, 683–712, doi:10.1007/s10579-019-09475-3.
8. Woodruff, A.G. (GIPSY) Georeferenced Information Processing System. *J. Am. Soc. Inf. Sci.* **1994**, *45*, 1–44.
9. Gritta, M. Where Are You Talking About? Advances and Challenges of Geographic Analysis of Text with Application to Disease Monitoring. Ph.D.; Thesis, University of Cambridge, Cambridge, UK, 2019.

10. Bo, A.; Peng, S.; Xinming, T.; Alimu, N. Spatio-temporal visualization system of news events based on GIS. In Proceedings of the IEEE 3rd International Conference on Communication Software and Networks, Xi'an, China, 27–29 May 2011; pp. 448–451, doi:10.1109/iccsn.2011.6014089.
11. Grover, C.; Tobin, R.; Byrne, K.; Woollard, M.; Reid, J.; Dunn, S.; Ball, J. Use of the Edinburgh geoparser for georeferencing digitized historical collections. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2010**, *368*, 3875–3889, doi:10.1098/rsta.2010.0149.
12. Leidner, J.L. Toponym resolution in text: Annotation, evaluation and applications of spatial grounding of place names, The University of Edinburgh, 2008.
13. Amitay, E.; Har'El, N.; Sivan, R.; Soffer, A.; Web-a-Where: Geotagging Web Content, in SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 273–280.
14. Karimzadeh, M.; Pezanowski, S.; MacEachren, A.M.; Wallgrün, J.O. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Trans. GIS* **2019**, *23*, 118–136, doi:10.1111/tgis.12510.
15. Gritta, M.; Pilehvar, M.T.; Collier, N. Which Melbourne? Augmenting Geocoding with Maps. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) **2018**, *1*, 1285–1296, doi:10.18653/v1/p18-1119.
16. D'Ignazio, C.; Bhargava, R.; Zuckerman, E.; Beck, L.; CLIFF-CLAVIN: Determining Geographic Focus for News Articles, *Proc. NewsKDD Data Sci. News Publ.*, 2014.
17. Lieberman, M.D.; Sperling, J.; Washington, D.C.; STEWARD: Architecture of a Spatio-Textual Search Engine, In Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, 2007, no. c.
18. LDC, ACE (Automatic Content Extraction) English Annotation Guidelines for Events V5.4.3 Linguistic Data Consortium, 2005, Available online: <https://www ldc.upenn.edu/collaborations/past-projects/ace>.
19. Dewandaru, A.; Supriana, S.I.; Akbar, S. Event-Oriented Map Extraction From Web News Portal: Binary Map Case Study on Diphteria Outbreak and Flood in Jakarta. *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)* **2018**, 72–77, doi:10.1109/icaicta.2018.8541345.
20. Ramage, D.; Hall, D.; Nallapati, R.; Manning, C.D. Labeled LDA. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1-EMNLP '09* **2009**, 248–256, doi:10.3115/1699510.1699543.
21. B.; Technologies, CLAVIN. Available online: <https://github.com/Novetta/CLAVIN>.
22. Teitler, B.E.; Lieberman, M.D.; Panozzo, D.; Sankaranarayanan, J.; Samet, H.; Sperling, J. NewsStand. *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems GIS '08* **2008**, 2008, 1,, doi:10.1145/1463434.1463458.
23. Andogah, G.; Bouma, G.; Nerbonne, J. Every document has a geographical scope. *Data Knowl. Eng.* **2012**, *81–82*, 1–20, doi:10.1016/j.datak.2012.07.002.
24. Li, H.; Srihari, R.K.; Niu, C.; Li, W. Location normalization for information extraction. *Proceedings of the 19th international conference on Computational linguistics*, 2002, pp. 1–7.
25. Srihari, R.K.; Li, W.; Cornell, T.; Niu, C. InfoXtract: A customizable intermediate level information extraction engine. *Nat. Lang. Eng.* **2006**, *14*, 33–69,, doi:10.1017/s1351324906004116.
26. P. A. Schrodtt and K.; Leetaru, GDEL: Global Data on Events, Location and Tone, 1979–2012, *Int. Stud. Assoc. Meet.*, pp. 1–49, 2013.
27. Leetaru, K.H. Fulltext Geocoding Versus Spatial Metadata for Large Text Archives: Towards a Geographically Enriched Wikipedia. *D-Lib Mag.* **2012**, *18*, 1–23, doi:10.1045/september2012-leetaru.
28. Lee, S.J.; Liu, H.; Ward, M.D. Lost in Space: Geolocation in Event Data. *Politi. Sci. Res. Methods* **2018**, *7*, 871–888, doi:10.1017/psrm.2018.23.
29. Handbook of Computational Approaches to Counterterrorism. *Handbook of Computational Approaches to Counterterrorism* **2013**, doi:10.1007/978-1-4614-5311-6.
30. Halterman, A., Massachusetts Institute of Technology Political Science Department Linking Events and Locations in Political Text Andrew Halterman, Massachusetts Institute of Technology, 2018.
31. Imani, M.B.; Chandra, S.; Ma, S.; Khan, L.; Thuraisingham, B. Focus location extraction from political news reports with bias correction. *2017 IEEE International Conference on Big Data (Big Data)* **2017**, 1956–1964, doi:10.1109/bigdata.2017.8258141.

32. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* **2014**, 1532–1543,, doi:10.3115/v1/d14-1162.
33. Halterman, A. Geolocating Political Events in Text. *Proceedings of the Third Workshop on Natural Language Processing and Computational Social Science* 2019, 29–39.
34. Yang, B.; Mitchell, T.M. Joint Extraction of Events and Entities within a Document Context. In *Proceedings of the Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Association for Computational Linguistics (ACL)*, 2016; pp. 289–299.
35. Leidner, J.L.; Lieberman, M.D. Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Spéc.* **2011**, 3, 5–11, doi:10.1145/2047296.2047298.
36. Kwok, K.L.; Deng, Q. GeoName. *the HLT-NAACL 2003 workshop* **2003**, doi:10.3115/1119394.1119398.
37. Morton-Owens, E.G., A Tool For Extracting And Indexing Spatio-Temporal Information From Biographical Articles in Wikipedia, 2012, Available online: http://www.cs.nyu.edu/web/Research/MsTheses/owens_emily.pdf.
38. Schilder, F.; Versley, Y.; Habel, C., Extracting spatial information: Grounding, classifying and linking spatial expressions, *Proc. Work. Geogr. Inf. Retr. SIGIR* 2004, pp. 1–3, 2004, Available online: http://publikationen.stub.uni-frankfurt.de/frontdoor/deliver/index/docId/9959/file/VERSLEY_Extracting_spatial_information.pdf.
39. Lan, R.; Adelfio, M.D.; Samet, H. Spatio-temporal disease tracking using news articles. *Proceedings of the Third ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health, HealthGIS* **2014**, 14, 31–38, doi:10.1145/2676629.2676637.
40. Monteiro, B.R.; Davis, C.A.; Fonseca, F. A survey on the geographic scope of textual documents. *Comput. Geosci.* **2016**, 96, 23–34, doi:10.1016/j.cageo.2016.07.017.
41. Bensalem, I.; Kholadi, M.-K. Toponym Disambiguation by Arborescent Relationships. *J. Comput. Sci.* **2010**, 6, 653–659, doi:10.3844/jcsp.2010.653.659.
42. Markert, K.; Nissim, M., Towards a corpus annotated for metonymies: The case of location names, *Proc. 3rd Int. Conf. Lang. Resour. Eval. Lr.* 2002, pp. 1385–1392, 2002.
43. Hogenboom, F. An Overview of Event Extraction from Text, *Comput. Sci.* **2011**.
44. Pustejovsky, J. et al., The Specification Language TimeML, pp. 1–15, 2004.
45. Wang, W.; Zhao, D.; Wang, N. Chinese News Event 5W1H Elements Extraction Using Semantic Role Labeling. *2010 Third International Symposium on Information Processing* **2010**, 484–489, doi:10.1109/isip.2010.112.
46. Khodra, M.L. Event extraction on Indonesian news article using multiclass categorization. *2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)* 2015, 1–5.
47. Rauch, E.; Bukatin, M.; Baker, K. A confidence-based framework for disambiguating geographic terms. *Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references*, 2003, 50–54.
48. Leidner, J.L.; Sinclair, G.; Webber, B. Grounding spatial named entities for information extraction and question answering. *the HLT-NAACL 2003 workshop* **2003**, doi:10.3115/1119394.1119399.
49. Habib, M.B.; Van Keulen, M. A Hybrid Approach for Robust Multilingual Toponym Extraction and Disambiguation. *Computer Vision* **2013**, 7912 LNCS, 1–15, doi:10.1007/978-3-642-38634-3_1.
50. Nissim, M.; Matheson, C.; Reid, J.; Recognizing Geographical Entities in Scottish Historical Documents., *Proc. Work. Geogr. Inf. Retr. SIGIR* 2004, 2004.
51. Adams, B.; McKenzie, G.; Gahegan, M. Frankenplace. In *Proceedings of the Proceedings of the 24th International Conference on World Wide Web 15 Companion; Association for Computing Machinery (ACM)*, 2015; pp. 12–22.
52. Buscaldi, D. Toponym Disambiguation in Information Retrieval. *Toponym Disambiguation in Information Retrieval* 2015.
53. Smith, D.A.; Crane, G. Disambiguating Geographic Names in a Historical Digital Library. *Computer Vision* **2001**, 2163, 127–136, doi:10.1007/3-540-44796-2_12.
54. Wei, W.W., University of Iowa Automated spatiotemporal and semantic information extraction for hazards, The University of Iowa, IA, USA, 2018.

55. Wang, J.; Zhang, J.; An, Y.; Lin, H.; Yang, Z.; Zhang, Y.; Sun, Y. Biomedical event trigger detection by dependency-based word embedding. *BMC Med. Genom.* **2016**, *9*, 45, doi:10.1186/s12920-016-0203-8.
56. Blei, D.M.; Carin, L.; Dunson, D.B. Probabilistic Topic Models. *IEEE Signal. Process. Mag.* **2010**, *27*, 55–65, doi:10.1109/msp.2010.938079.
57. Řehůřek, R. Scalability of Semantic Analysis in Natural Language Processing, p. 147, 2011, Available online: http://radimrehurek.com/phd_rehurek.pdf.
58. M., D.; Blei, A.Y.N.; Jordan, M.I., Latent Dirichlet Allocation, *J. Mach. Learn. Res.*, **2003**, *3*, pp. 993–1022.
59. Papanikolaou, Y.; Tsoumakas, G. Subset Labeled LDA for Large-Scale Multi-Label Classification 2017.
60. Kang, D.; Park, Y.; Chari, S.N. Hetero-Labeled LDA: A Partially Supervised Topic Model with Heterogeneous Labels. *Public-Key Cryptography – PKC 2018* **2014**, *1*, 640–655, doi:10.1007/978-3-662-44848-9_41.
61. Greene, D.; O’Callaghan, D.; Cunningham, P. How Many Topics? Stability Analysis for Topic Models. *Public-Key Cryptography – PKC 2018* **2014**, *1*, 498–513, doi:10.1007/978-3-662-44848-9_32.
62. 10.1162/153244303322533223. *Appl. Phys. Lett.* **2000**, *1*, 39, doi:10.1162/153244303322533223.
63. Leidner, J.L. An evaluation dataset for the toponym resolution task. *Comput. Environ. Urban. Syst.* **2006**, *30*, 400–417, doi:10.1016/j.compenvurbsys.2005.07.003.
64. Gritta, M.; Pilehvar, M.T.; Limsopatham, N.; Collier, N. What’s missing in geographical parsing? *Lang. Resour. Evaluation* **2018**, *52*, 603–623, doi:10.1007/s10579-017-9385-8.
65. Ha, L.Q.; Hanna, P.; Ming, J.; Smith, F.J. Extending Zipf’s law to n-grams for large corpora. *Artif. Intell. Rev.* **2009**, *32*, 101–113, doi:10.1007/s10462-009-9135-4.
66. Dewandaru, A. Event Geoparsing Indonesian News Dataset, IEEE Dataport, 2020. .
67. Bender, E.M.; Lascarides, A. Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from Semantics and Pragmatics. *Synth. Lect. Hum. Lang. Technol.* **2019**, *12*, 1–268, doi:10.2200/s00935ed1v02y201907hlt043.
68. C. E.; Data, PETRARCH : The successor to TABARI, no. August 2014, pp. 1–3, 2019.
69. GADM database of Global Administrative Areas, version 2.0, Berkeley, CA Univ. Berkeley, 2012.
70. Purwarianti, A.; Andhika, A.; Wicaksono, A.F.; Afif, I.; Ferdian, F. InaNLP: Indonesia natural language processing toolkit, case study: Complaint tweet classification. *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)* **2017**, 1–5, doi:10.1109/icaicta.2016.7803103.
71. Strohmeier, D.; Eggers, T.; Haupt, M. Waverider Aerodynamics and Preliminary Design for Two-Stage-to-Orbit Missions, Part 1. *J. Spacecr. Rocket.* **1998**, *35*, 450–458, doi:10.2514/2.3375.
72. Murtaugh, M.A.; Gibson, B.S.; Redd, D.; Zeng-Treitler, Q. Regular expression-based learning to extract bodyweight values from clinical notes. *J. Biomed. Informatics* **2015**, *54*, 186–190, doi:10.1016/j.jbi.2015.02.009.
73. Yang, J.; Zhang, Y. NCRF ++ : An Open-source Neural Sequence Labeling Toolkit. In Proceedings of the Proceedings of ACL 2018, System Demonstrations; Association for Computational Linguistics (ACL), 2018; pp. 74–79.
74. Lin, J.C.-W.; Shao, Y.; Zhang, J.; Yun, U. Enhanced sequence labeling based on latent variable conditional random fields. *Neurocomputing* **2020**, *403*, 431–440, doi:10.1016/j.neucom.2020.04.102.
75. D.; Mimno, H.M.; Wallach, E.; Talley, M. Leenders, and A. McCallum, Optimizing Semantic Coherence in Topic Models, Proc. 2011 Conf. Empir. Methods Nat. Lang. Process., no. 2, pp. 262–272, 2011, Available online: <http://mimno.infosci.cornell.edu/papers/mimno-semantic-emnlp.pdf>.
76. Mimno, D. Package ‘mallet,’ Compr. R Arch. Netw., pp. 1–11, 2015, Available online: <https://cran.r-project.org/web/packages/mallet/mallet.pdf>.
77. Řehůřek, R.; Petr, S. Software Framework for Topic Modelling with Large Corpora. ELRA, p. 45, 2010.
78. Q.; Li, H.J.; L Huang, Joint Event Extraction via Structured Prediction with Global Features, 2013.
79. D. McClosky, M. Surdeanu, and C. D.; Manning, Event extraction as dependency parsing, ACL-HLT 2011-Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol., vol. 1, pp. 1626–1635, 2011.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).