

Article

Event Geoparser with Pseudo-Location Entity Identification and Numerical Extraction in Indonesian News Corpus

Agung Dewandaru^{1,*}, Dwi Hendratmo Widyantoro¹ and Saiful Akbar²

1 School of Electrical and Informatics Engineering, Institut Teknologi Bandung, Indonesia

2 PUI-PT Artificial Intelligence for Vision, Natural Language Processing & Big Data Analytics [AI-VLB]

* Correspondence: dewandaru@gmail.com.

Abstract: One of the most important component of a Geographic Information Retrieval (GIR) is the geoparser, which performs toponym recognition, disambiguation, and geographic coordinate resolution from unstructured text domain. However, news articles which report several events across many place references mentioned in the document is not yet adequately modeled by regular geoparser types where the scope of resolution is either on toponym-level or document-level. The capacity to detect multiple events, geolocate its true locations and coordinates along with their numerical arguments are still missing from modern geoparsers, much less in Indonesian news corpora domain. We propose a novel type event geoparser which integrates an ACE-based event extraction model and provides precise event-level scope resolution. The geoparser casts the geotagging and event extraction as sequence labeling and uses Conditional Random Field with keywords feature obtained using Aggregated Topic Model as a semantic exploration from large corpus, which eventually increases the generalizability of the model. The geoparser also use Smallest Administrative Level feature along with Spatial Minimality-derived algorithm to improve the identification of Pseudo-location entities, resulting 19.4% increase for weighted F1 score. As a side effect of event extraction, the geoparser also extracts various numerical arguments and able to generate thematic choropleth map from a single news story.

Keywords: geoparser, geographic information retrieval, event extraction, argument extraction, information extraction, named entity recognition, conditional random function, 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 these information embed geographical references, directly or indirectly in forms of toponyms (*place names entities*) or its references. One estimate stated at least 20 percent of Web pages include recognizable geographic identifiers [1] that is mainly present in unstructured form. Hence, numerous types of Geographical Information Retrieval (GIR) models, method and prototypes have been developed with the aim of extracting and retrieving 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 database, GIR systems typically have to extract and infer geographic location or coordinates from many types of noisy information and ambiguities that is prevalent in the unstructured natural language form. Thus, GIR usually starts with the geoparser

component to extract geographic information from text, which is then followed by some indexing and retrieval mechanisms typically in form of maps further down the pipeline. The geoparsing process is composed of two tasks [7]: 1) *geotagging*, i.e. detecting geographical references or toponyms from text and 2) *geocoding*, which means to resolve these into precise coordinates via some means of disambiguation methods. The result will be further processed by GIR application to infer associations between various 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 has been made in the field of geoparsing, from Woodruff who introduced the first geoparsing prototype within GIPSY in 1994 [8] to as recent as 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 as *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 *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 event-level resolution scope geoparsers (or *event geoparsers* for short) needs to be capable of: 1) detecting what event presented in the document and 2) infer the precise location of the event reported (event geolocation) from the detected toponyms in the document. Additionally, 3) event geoparser should be able to discover which event argument(s) (especially NUMEX or any

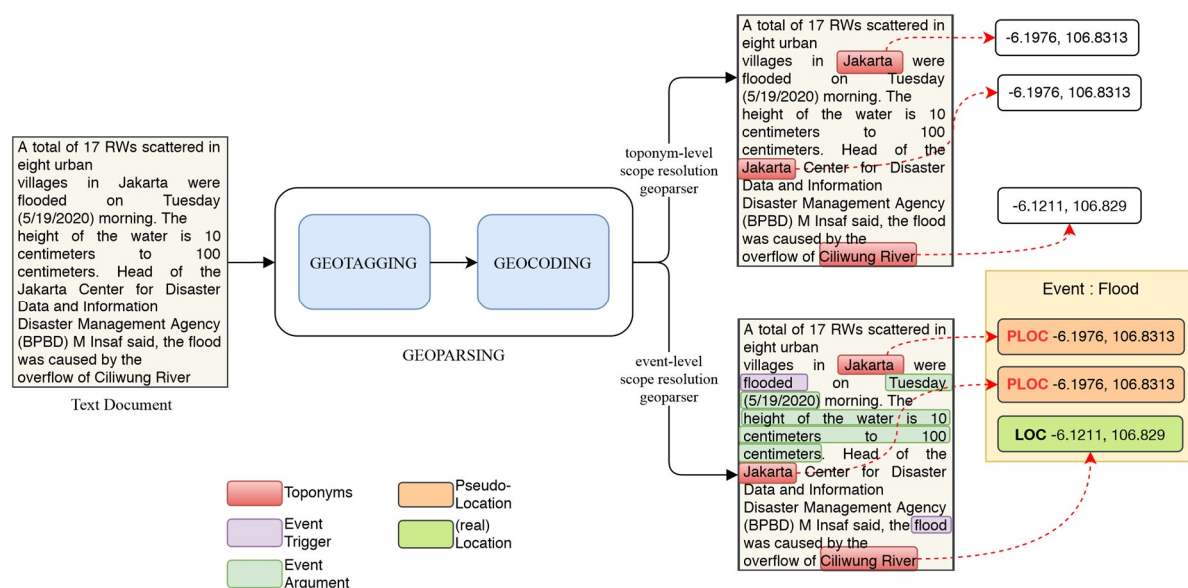


Figure 1 A comparison of toponym-level and event-level scope of resolution. Unlike toponym-level geoparser, the event geoparser is not only detecting toponym and resolve the coordinate, but also detecting what event(s) happened and infer which toponyms is the real location (LOC) or only pseudo-location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

string convertible to numeric value) associated with the detected event(s) for 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, which we argue would need an event geoparser on its own.

This paper presents an novel implementation of an event geoparser that is based on ACE event model [18] which tightly integrates event extraction and the toponym resolution which usually dealt

separately. The model decomposes an event into its trigger (or anchor), related entities, resolved (grounded) locations, and semantic roles that it may have and its arguments, especially numerical ones. The geoparser model cast the geotagging and event extraction as sequence labeling task, hence it uses Conditional Random Field (CRF) sequence labeler as a statistical method on the 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 was described in our earlier work [19] (which we will later identify as *large corpus*) and 83 news articles composed of 926 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 *small corpus* in which the event geoparser model is mainly trained from. The geoparser also use *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 will be proved to be very important in the ability of the geoparser to detect the pseudo-location, which is necessary for geolocating event.

To improve the model generalizability when tested on unseen data, we propose a semantic 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. This topic model motivation is to efficiently exploit a large number (44,280) of unique news tags as the labels offered by the 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 either as a component for *event keywords* feature or *regular expression* feature to help improve geoparser's performance.

2. Related Works

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

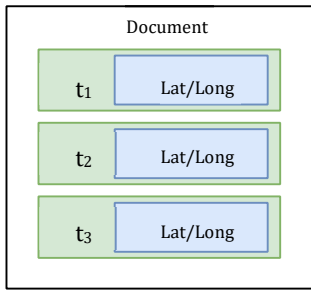
The majority of geoparsers works on either *toponym-level resolution scope* or *document-level resolution scope*. Toponym-level resolution scope means that it works with an assumption that every toponym will have assigned coordinates, typically via some disambiguation and resolution process (grounding) 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 work with this assumption. On the other hand, geoparsers which have document-level resolution, are set out to finding 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.

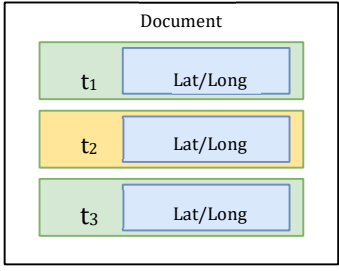
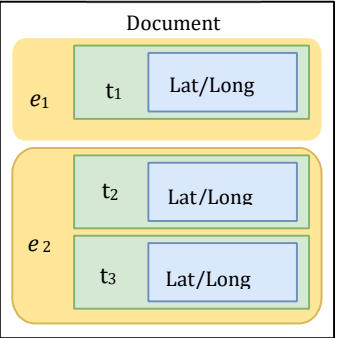
The last type and the most recent development is what we call as *event-level resolution scope* geoparser. It will try to detect event(s) within document and to resolve the location or geographic scope of those events. There are only very few geoparsers which has this capability, and they are still very limited. Most event geoparser 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; followed by event detection (or often referred as *event coding*) step; and ended 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]. Because of this independency of the geoparser module, the integration of event coding and geoparser is typically done in a black box approach: the geoparser does not know anything about the event structure or semantics; and the event coding system simply attach 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 geoparser. 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[29] and Profile[30] 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 document which has more than one event, or an event which has several locations. Both of which are common within our corpus and other 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 with its locations. Mordecai models a 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 it 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 (and its trigger) 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 POS tags, pretrained GloVe [31], dependency label, and signed distance of word from the anchor [32]. Even though it is effective on the narrow, political dataset that Mordecai 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 describe in joint event extraction model in [33]. On 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 = toponym, D = set of words in the document, \mathbb{E} = set of events in the Document, \mathbb{G} = set of resolved coordinate/footprint | Example Geoparser/GIR |
|-------------------------------------|---|--|
| Toponym-level (geocoded toponym) |  <p>Output is geographical coordinate for each toponym in the document based on recognized toponyms entities within the document.</p> $g = \{ \text{resolve}(t, D) \in \mathbb{G} \mid \forall t \in D \}$ | SPIRIT [6] Edinburgh Geoparser [11] Spatial Minimality[12] CLAVIN [21] Camcoder [15] |

| | | |
|--|---|---|
| Document-Level (geographical scope of document) |  <p>Output is single geographic coordinate or scope of the document based on some scoring function of recognized toponyms within the document.</p> $g \in \mathbb{G} = \text{resolve}(\arg\max_{t \in D} \text{focus_score}(t, D))$ | Web-a-Where[13] NewsStand ¹ [22] GeoTxt [14] Mahali [23] CLIFF [16] |
| Event-Level (geolocated event) |  <p>Output is geographical coordinate for each locations of recognized event(s) within the document.</p> $g = \left\{ \text{resolve}(e) \in \mathbb{G} \mid \forall e \in \mathbb{E}, t \in D, \text{is_location}(t, e) \right\}$ | Mordecai [29] Profile [30] GDELT 1 (TABARI + Leetaru [26]) Petrarch + Mordecai [29] 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 referring to names of a places. This process is commonly referred as *geotagging* or *toponym recognition*². Geotagging requires methods for discriminating location entities of place names (*toponyms*) from other entities. The dominant geotagging method used in most geoparser is to incorporate *gazetteer lookup*, which is a lookup process from external resource of place names and basic geographic information for simple string matching. Generally, the matching toponym string (which may consists of several tokens) inside the gazetteer indicates strong probability of such token being place names, with some exceptions need to be made 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 varies 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 [34]. We call an ontological gazetteer that maintains correct hierarchy for all its entries as a *strict gazetteer*. GADM, for example, can be considered as strict gazetteer with four level of administrations from total 368.735 administrative areas. Geonames [35] is an ontological gazetteer with 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) has been placed

¹ Hybrid geoparser that offer document level on top of toponym-level scope of resolution.

² Some authors referred *toponym recognition* process as *geoparsing*, such as [39], [75]. We will stick to the more recent trend of interpreting geoparsing to include both toponym recognition and resolution as in [7] and many others.

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 [36]. Data driven approach require an annotated corpus (often annotated using BIO scheme) which 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 [37], word forms or capitalization. It means that not all match 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 [29]), LingPipe (used in [23], [38]), GATE ANNIE ([4], [14]), Spacy [14], Stanford CoreNLP (used in [16], [21]), NCRF++ (used in [9]) and others. Most of these NER in turn are using 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].

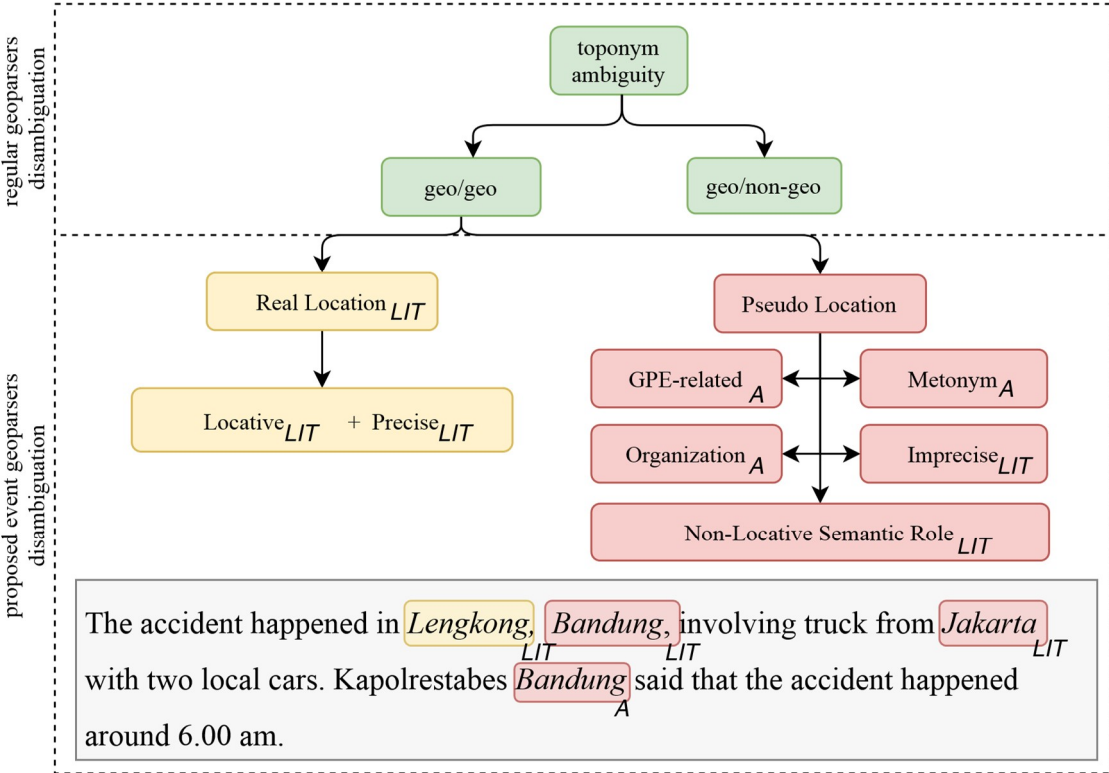
Generally, both gazetteer and NER approach have been successfully used by geoparsers to tag and extract the toponyms the text in the geotagging step. However, the main challenge here is that the extracted toponyms do not necessarily indicate the location of events mentioned in the news document. Furthermore,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

2.3. Geotagging 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 one. Literal toponym carries notion of location where some event happens. On the other hand, associative toponyms indirectly carry the notion of location. While literal toponyms seems to be a major use case, it is actually only comprising 53.5% in his evaluation on GeoWebNews corpus [7], with the rest of the use are 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 is when a toponym 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 referents 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], and [28]. 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) 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 one). Thus, we will extend the dichotomy to accommodate 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), it 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 handled well by existing geoparsers.

Figure 2 Taxonomy of Toponym Ambiguity. Even though regular geoparsers already capable of filtering geo/non-geo ambiguities and assign disambiguated coordinate to geo/geo referential ambiguities, they cannot yet handle event geolocation properly, i.e. recognize and resolve toponyms that is both locative and precise of *particular event* by discarding all “pseudo-location”



LIT: Literal toponym A: associative toponym
entities which is irrelevant to that event. Note that pseudo-location entity 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 [41]; 3) as evidonym (part) of organization name [8]. 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 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, hence it is considered as pseudo-location entities 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 contains a more precise toponym. For example, in this sentence, “A 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 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 resource (e.g. gazetteer) is not enough (as in [6]) as toponym mentions in a single document do not always mean to where the event had happened. These toponyms may appear in lots of various sentence contexts in various syntactical patterns which 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 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 that in that sentence that all toponyms there are literal. Lengkong, Bandung, Jakarta are all literals toponyms. Hence there is obviously a substantial need for discriminating the locative toponym and not. 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. Also, regular geoparsers or NER are not equipped with event semantics to differentiate locative vs literal toponym, 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 label). Regular geoparsers (such as CLIFF-CLAVIN, Edinburgh Geoparser) is able to detect (tag) those toponyms without any issues. However, even though those toponyms are all literal toponyms (Jakarta, Bandung, and Lengkong) but 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) associated with the event, and later on correctly infer the real location entities where the event located (i.e. finding locative and precise toponym) and its correct coordinate 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 particular street, road segment or coordinate, while an earthquake event may easily span across province or even 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 [42]. 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 [43]. 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 model the temporal aspects of an event and less about spatial and grouping aspect of 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 [44] and [45]. However, both models are not suitable to group various roles (especially 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 [33], 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 assumption to model the events. Therefore, we chose to base on the ACE model loosely that 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 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 arguments slots except fixed spatial, temporal, and semantic keyword entities.

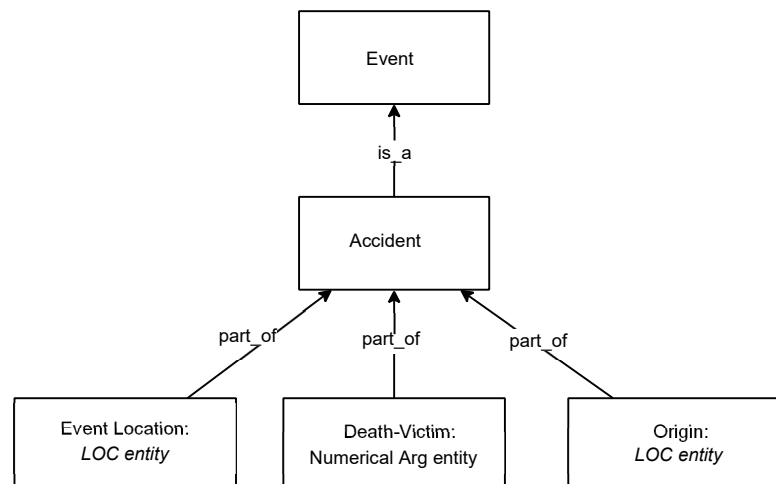


Figure 3 Sample Ontology of Vehicle Accident. Events can be modeled as grouping of various semantic roles and its arguments, forming templates for different type of event. For example, the Accident event has two semantic roles with regard to location entity and one numerical argument.

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 emphasize on 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 [33]. 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 persons and 20 persons). Within the context of geoparsing, the extracted event types and its arguments may provide additional data to the event geolocation process for better inference, while the extracted arguments may provide a richer context for the generation of 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 do not discuss coordinate-level accuracy while geoparser aim 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 phases where toponym-level geoparser works with raw text (without information of any event structure) and the output is attached to the event extraction result.

Thus, the objectives we believe still missing in state-of-the-art geoparser field is 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 instance of event(s) described in document and in which precise location that such event happened. Event extraction framework will provide event labels and arguments which will provide semantic event context (in which the inferred location data is associated), which eventually would improve the performance of the geoparser; the numerical arguments extracted along with the event would 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 about toponym resolution step which typically starts after toponym recognition. Toponym resolution sought to resolve referential ambiguities the literal toponym detected in the 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 represents toponym resolution insights that is coded into the system as simple rules or simplifying assumptions. For example, the *population heuristics*, prefers higher population referent to lower population referent candidate [46]. Thus, if ambiguous places are present, the system will resolve it or prefer to the most populated place. This is used in the work of [13], [11], 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 document [16]. Similar to that, there is also very common “*one sense per discourse*” heuristic, which assign only one interpretation across several instances of the same toponym, used in [13], [47]. Another heuristic used in the context of document-level geoparser is that of *frequency heuristic*: the geoparser prefer interpretation whose number of occurrences of the toponyms is highest within the document. The more it appears; 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 [48] or classification method [49]. 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 on the next section.

Lastly, toponym resolution strategies often make use of the map information available to prefer lesser place distance or *geographic proximity* [49], or overlapping areas [8], [50]. 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 [51] and [46]. This strategy is introduced in [52] 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 premise (called *geometric minimality heuristic*) that the correct place candidates is composing the smallest region that is able to contain the whole set of toponyms inside a document [12].

Generally, these heuristics depends 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 phase.

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 [53]. This keyword, if obtained from 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 [54]. Topic modeling is often used as an automatic statistical method of dimensionality reduction for clustering articles into a set of topics [55] which itself is a distribution of related keywords. Thus topic modeling is a good approach to the semantic relatedness concept [56]. The output of topic models is typically a set of topic clusters, each of which is essentially probability distribution over words. This would provide cluster of related terms which resembles notion of topic relations between the words. The “top-words” are 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 existence of event trigger words.

Topic modeling models typically trained as unsupervised learning fashion, with the main input is the number of topics that it should produce from the training session. For example, in the Latent Dirichlet Allocation (LDA) model the main parameter is K (number of) topics [57]. 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 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], [58], [59]. With the introduction of tags as label, the unsupervised nature of LDA becomes supervised in Labeled LDA.

One of the LDA derived model that use document tags is the Labeled LDA [20], whereby it puts one – to – one correspondence constraint between document tag and latent topic. A topic has a string label (caption) taken from document tag that can be used for further inference. Unlike the unsupervised LDA, Labeled LDA (LLDA) incorporate 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 often the case in topic modeling frameworks: there is no clear-cut method to specify the number of clusters of the topics [60]. 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 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 [61] to assist the widely used conditional random field (CRF) sequence labeler to provide better accuracy of the event trigger classification.

3. Geospatial News Event Extraction Corpus

The objective of this corpus is to be the material of experiment whether we can gain improvement by integrating event extraction framework into the geoparsing. To the best of our knowledge, there is yet any news corpus that provide both the correct geographical disambiguation as well as event extraction labels that is suited to training and testing, much less one in Bahasa Indonesia. The criteria that we look for the news dataset was : 1) it covers major geospatial events; 2) it has resolved all place names to the correct coordinate and administrative entities; and 3) it has 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[62], Wiktor and GeoWebNews [63] provided geoparsed corpus but it does 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 open dataset and we are not able to access it. In Indonesian context, there is the 5W1H-style news extraction and corpus [45], but without geoparsed toponyms and detailed event semantics. With these circumstances, we decided to contribute by constructing one in Bahasa Indonesia.

We first use the corpus of our earlier work [19] which consisted of 13 years of news articles (2005-2018), totaling 645.679 articles with 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 average 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.

Secondly, we select a random subset of the corpus of 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*). To achieve this, four annotators have further annotated 926 sentences from 83 articles from the subset of detik.com, Kompas.com and CNNIndonesia.com. The annotations are done for each token following the BIO-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 entity types annotation, event annotation, geospatial disambiguation annotation, and Pseudo-location tags.

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 event 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 & Examples |
|-----|---|
| EVE | <p>Description: Event Triggers: word(s) that indicate an event has occurred. Examples:</p> <ol style="list-style-type: none"> 1. Flood happened in ... (<i>Banjir terjadi di ...</i>) 2. ...that the fireworks from the band triggered fast-moving fire flame. (<i>... bahwa kembang api dari band, memicu kobaran api yang bergerak cepat.</i>) |
| ARG | <p>Description: Non-named arguments related to event. May include numerical or non-numerical arguments.</p> <p>I. Event Arguments for Flood</p> <ol style="list-style-type: none"> 1. <u>Height of flood: Height-Arg</u> The height of the water reached 2 meters ... (<i>Ketinggian air yang mencapai 2 meter...</i>) 2. <u>Number of Victim (Deaths): DeathVictim-Arg</u> At least 41 people killed due to the flood. (<i>Sedikitnya 41 orang tewas akibat banjir ini.</i>) 3. <u>Number of Evacuee: Evacuee-Arg</u> Indonesian Field Hospital handled 9 victims and 346 evacuee. (<i>Rumah Sakit Indonesia di Nepal Tangani 9 Korban dan Tampung 346 Pengungsi</i>) 4. <u>Number of Affected houses: AffectedHouse-Arg</u> Flooding caused 4,991 houses to be submerged... (<i>Banjir menyebabkan sekitar 4.991 rumah terendam ...</i>) <p>II. Event Arguments for Quake:</p> <ol style="list-style-type: none"> 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. Quake Center: Central-Arg
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. Quake Depth: Depth-Arg
The depth of the earthquake was **10 kilometers**.
(*Kedalaman gempa 10 kilometer*)
-

III. Event Arguments for Fire:

1. Number of houses burnt: HouseBurnt-Arg
A house at Mampang Prapatan burned...
(*Sebuah rumah di Mampang Prapatan Terbakar*)
 2. How many fire hotspots: Point-Arg
There are **nine fire spots...**
(ada **sembilan titik api...**)
 3. Units of fire truck dispatched: DispatchedTrucks-Arg
...12 firetrucks were dispatched.
(*...12 damkar dikerahkan*)
-

III. Event Arguments for Accident:

1. License plates: Plate-Arg
... which has **B 9667 ZX** license plate.
(...beropol **B 9667 ZX**)
 2. Vehicle type involved: Vehicle-Arg
...hit hard on the back of the **truck**.
(*menghantam keras bagian belakang truk*)
 3. The length of jam: Length-Arg
The accident caused up to **2 kilometers** traffic jam.
(*kecelakaan mengakibatkan kepadataan kendaraan hingga 2 kilometer*)
 4. Origin or Destination: FromTo-Arg
The accident caused up to **2 kilometers** traffic jam.
(*kecelakaan mengakibatkan kepadataan kendaraan hingga 2 kilometer*)
-

IV. Others (may appear in more than one events above):

1. Numerical Monetary loss: MonetaryLoss-Arg
The loss is estimated around **hundreds of millions rupiahs**. (*Kerugian diperkirakan mencapai ratusan juta rupiah.*)
 2. Time or Date of event: Time-Arg
- ...at **15.25 WIB**.
(*...pukul 15.25 WIB*).
- ... as reported by by AFP news agency on Friday (**3/25/2011**)
(*...dilansir kantor berita AFP, Jumat (25/3/2011)*)
 3. Cause of event: Cause-Arg
Example: ...caused by **River Kuncir** overflow.
(*...disebabkan luapan air Sungai Kuncir*)
 4. Affected families: AffectedFamily-Arg
...which caused **935** families to be affected.
(*...menyebabkan 935 KK terdampak*)
 5. Street names: Street-Arg
There is a fire near **Street KS Tubun Raya** behind the Bimo Hotel
(*ada kebakaran di dekat Jl KS Tubun Raya belakang hotel Bimo*)
-

ORG Organization (such as military or civilians)
Rank/Positions within it (*pangkat/jabatan*)

Useful for classifying Pseudo LOCs.

Example: **BPBD and TNI...**

(*BPBD bersama TNI...*)

Governor of East Java ...

(*Gubernur Jawa Timur...*)

LOC 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.*)

O Other Entities

The next set of annotation is 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 arguments (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 provide the list of disambiguation options along with the approximate central coordinate (centroid) of that geographical feature. Most of these LOC tags are in form of Geo-Political Entities (GPE) definition of ACE so it is desirable to use an administrative-based gazetteer to reference it. Also, there appear to be a recurring pattern of specifying toponyms in a consecutive and hierarchical manner, starting from the lower level to higher level (e.g.: from village, up to the province level).

Among the open data gazetteer that is 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 is not as comprehensive as Geonames, it is more rigorously structured in a sense that every upper administrative area is always composed of smaller elements. This would help the heuristic of containment that we have discussed earlier. 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 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 uses BRAT tool to annotate the corpus, later on it is 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 assign Pseudo-location Entities to be a precise location entities (toponym) which is inhabited place name 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 with regard to a huge area-related event such as Earthquake, 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 kilometers 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 regions. 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*[64] with the following label statistics of entities, events, and arguments (Table 3).

Table 3 Label Statistics within Event Geoparsing Indonesian News Dataset

| Type | Label | Count | Type | Label | Count |
|--------|-------|-------|----------|---------------------|-------|
| Entity | B-LOC | 302 | Argument | Duration-Arg | 31 |
| | I-LOC | 128 | | Affectedvehicle-Arg | 14 |

| | | | | |
|----------|-----------------|------|----------------------------|-----|
| | B-PLOC | 466 | AffectedHouse-Arg | 48 |
| | I-PLOC | 185 | AffectedCities-Arg | 2 |
| | B-EVE | 597 | AffectedFacility-Arg | 8 |
| | I-EVE | 103 | AffectedFamily-Arg | 20 |
| | B-ARG | 962 | AffectedField-Arg | 9 |
| | I-ARG | 1048 | AffectedPeople-Arg | 20 |
| | B-ORG | 288 | AffectedInfrastructure-Arg | 14 |
| | I-ORG | 285 | AffectedHouse-Arg | 48 |
| Event | ACCIDENT-EVENT | 130 | AffectedVillage-Arg | 32 |
| | FLOOD-EVENT | 195 | AffectedRT-Arg | 4 |
| | QUAKE-EVENT | 127 | AffectedFacility-Arg | 8 |
| | FIRE-EVENT | 179 | Time-Arg | 422 |
| | LANDSLIDE-EVENT | 18 | Published-Arg | 79 |
| | MEETING-EVENT | 4 | Reporter-Arg | 39 |
| Argument | JAM-EVENT | 16 | Evacuee-Arg | 20 |
| | Vehicle-Arg | 126 | Spot-Arg | 9 |
| | Hospital-Arg | 47 | DeathVictim-Arg | 198 |
| | Place-Arg | 1044 | WoundVictim-Arg | 24 |
| | Street-Arg | 157 | MonetaryLoss-Arg | 3 |
| | Cause-Arg | 52 | OfficerOfficial-Arg | 397 |
| | To-Arg | 32 | Depth-Arg | 29 |
| | From-Arg | 16 | Repeat-Arg | 2 |
| | Plate-Arg | 59 | Central-Arg | 89 |
| | | | | |
| | | | | |
| | | | | |

4. Approach

4.1. Task Formulation

As noted in section 2, we are going to modify 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 :

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

where A_q is a q -th element from set of all entities types, $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$. 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 prefix applied to each types indicate its position as beginning of entities or inside it. Similarly, event trigger type (t) and semantic role label type (r) is expressed as:

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

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

where \mathbf{T} is set of all event trigger labels such as "FLOOD-EVENT", "QUAKE-EVENT" and \mathbf{R} is set of all semantic role labels like "Height-Arg", "DeathVictim-Arg", etc. (Please refer to Table 3 for all possible labels for semantic roles and event types). Next and one of the perhaps one of most

important variable is the pseudo-location labels which subcategorize LOC entities into either Pseudo-location (PLOC) or real location (LOC):

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

where \mathbf{P} is only composed of either "PLOC" or "LOC". Note that we do not require the use verb as anchor word. Instead, it may be single multi-word non-verb entities that is deemed relevant [65].

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 labels in the document. Instead it represents the geographic coordinate of true location(s) of the event, hence the domain is geographic. It is possible that an event has several true locations, e.g. quake event can easily span multiple places or cities reported by the news article. Thus, the set of true location(s) are obtained by the process of resolving toponyms of the remaining location entities after discarding the pseudo-location entities associated with the event.

These set 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 instance of the same type found within an instance, then it is likely that it needs to be coreferenced 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. Architectural View of Event Geoparsing

This section will describe our architectural, systematic approach for integrating geoparsing with event extraction, 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 take place. We extend the regular workflow description of GIR and geoparsing process following [63] and generalized from our discussion from earlier section, by combining an event extraction step to extract the event, and the event's arguments (see Figure 4 below).

The two steps inside our standard *toponym-level geoparsing phase* are:

1. Geotagging, in which named literal geographical entities (toponyms) are recognized from other named entities. This is where the Named Entity Recognition typically invoked to recognize location entities. In our model, we do not invoke any external Named Entity Recognizer. Instead, the function is fulfilled by entity extraction step, which recognize and classify entities into event trigger (EVE), unnamed event argument (ARG), named entity of organization (ORG) along with location (LOC) entities, the latter of which is the toponym target.
2. Geocoding (or toponym resolution) in which correct toponyms are disambiguated from other toponym candidates (potential referents) and then assigned correct geographic coordinate. This is done per toponym to have toponym-level scope resolution.

We are hoping to have a deeper integration of event extraction and geoparsing by extending those original steps. In particular, we use event extraction phase in after geotagging and geocoding. This will provide event record data to be stored along with place data. So, the next two steps is the *event extraction phase*, which is comprised of:

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

The final phase is to resolve the location of the event (*event-level geoparsing*). This phase 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 4. It starts with geotagging step, which involves cleaning, sentence splitting and tokenization of the small corpus. 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 with toponym detection but does not necessarily leads to a detected toponym. We are using Global Administrative Areas (GADM) database (Global Administrative Areas, UC Berkeley 2012) 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 present in GADM database. The county where Prescott 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 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. We use InaNLP [67] 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 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 is 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 are LOC tags (for each detected toponym) along with EVE (event trigger), ARG (event arguments, can be numerical or string), and ORG (named entity of organization), each having B and I, indicating beginning or inside the token respectively. The output of this step is then carried forward to the event trigger classification step.

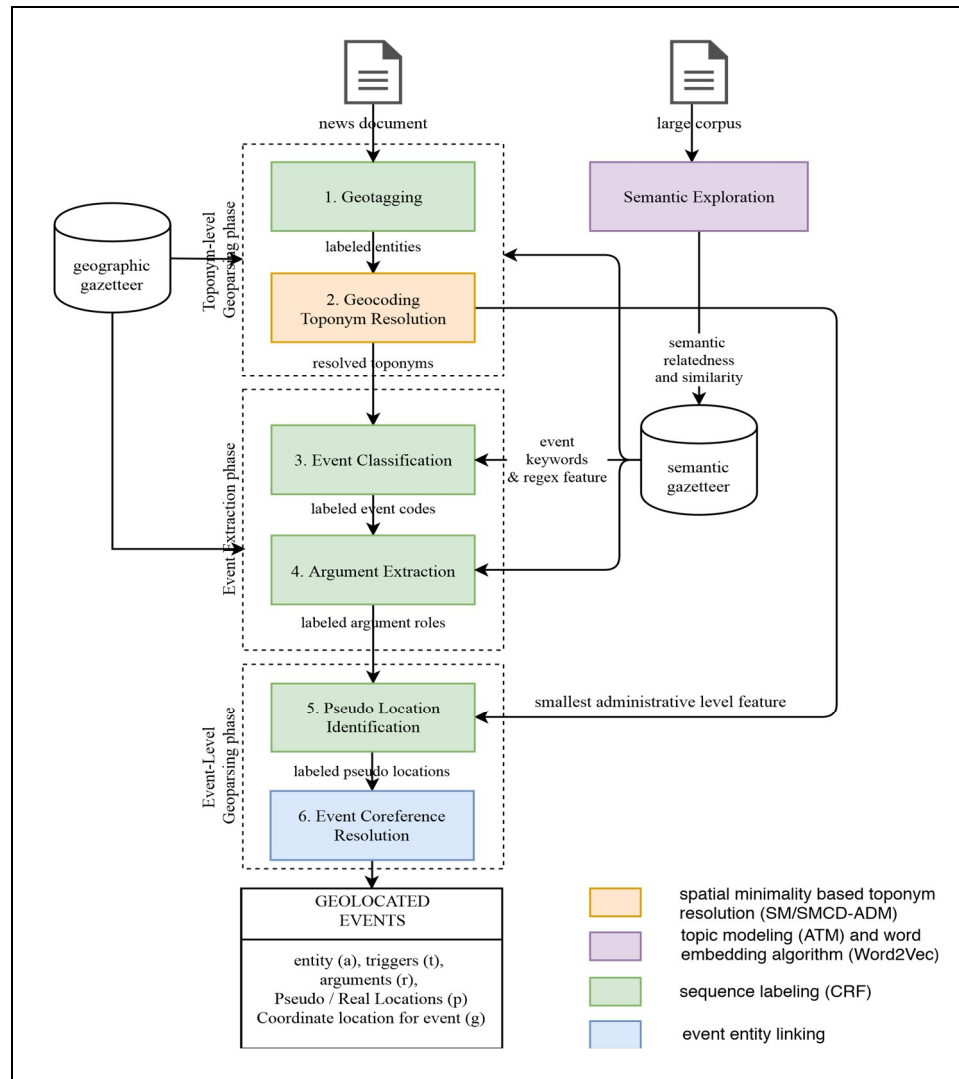


Figure 4 Integrated Event Extraction and Geoparsing accept news document as input, resolving toponym and other entities (a), event triggers type (t), arguments (r), and event locations (g) from text. It is chaining geotagging with toponym resolution and event extraction. The system use semantic gazetteer for features and regular expression rules learned from large corpus to increase the precision and recall and geoparser accuracy.

The event trigger classification step is then commenced with entity features that has been extracted from earlier step, notably the EVE entities. 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 where each argument types (e.g. DeathVictim-Arg) is inferred for each ARG entities.

The next step is Pseudo-location Detection, where every LOC entities are classified either as true location or pseudo-location one. The Pseudo-location tags is also fit and tested using the results coming from earlier steps. However, as an additional feature we use *smallest administrative level* (SAL) feature to check whether a location entity is smallest administrative level or not. This needs result from the disambiguation and toponym resolution which uses geographic gazetteer. In our case, we use spatial minimality framework explained in the later section (section 4.5). Note that these steps which involves CRF would require initial training first by fitted towards the training set. This will be discussed on the next Result section.

Table 4 Features for Entity, Event, Argument, and Pseudo-location Identification

| Category | Type & Source | Features |
|-------------------------------------|---|---|
| Event Keywords | Semantic Relatedness from ATM | Binary feature Is_Event_Keyword(w): whether a word is included in shortlisted trigger word or not. Composed of four (geospatial) events word bigrams list : flood, quake, fire, and accidents. |
| Smallest Administrative Level (SAL) | Geographical Feature | Is_SAL(t): whether a mentioned toponym has smallest administrative level in the document |
| Gazetteer Detection | Geographical Resource: GADM database + US_Cities | Is_Toponym(w): a word is listed in Hierarchical Gazetteer or not. |
| 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 |
| 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 | 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 | Output labels from entity extraction step | B-LOC, B-EVE, B-ARG, B-ORG I-LOC, I-EVE, I-ARG, I-ORG |
| Event | Output labels from event trigger classification step | FLOOD-EVENT, FIRE-EVENT, QUAKE-EVENT, ACCIDENT, EVENT |
| Argument | 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 document, our corpus offered a vast topic space [19], and we are mostly interested in the different types of geospatial events with its detailed attributes. As in every text documents, there can be a lot of topics discussed in the news articles, each topic can have a typical characteristics: 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 models and its (mostly) unsupervised learning algorithms. LDA is the prominent and simple topic model which has grown to 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 is taken from tags of each document in the corpus. The objective for ATM is to provide topic modeling tool while also solves 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, on which case can be considered as an extreme multi-label classification problem [58].

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 need to be taken as having too many number of topics will typically result in over clustering topics into a small and highly similar clusters [60], hence one important element of ATM is the merging of topics which has same labels.

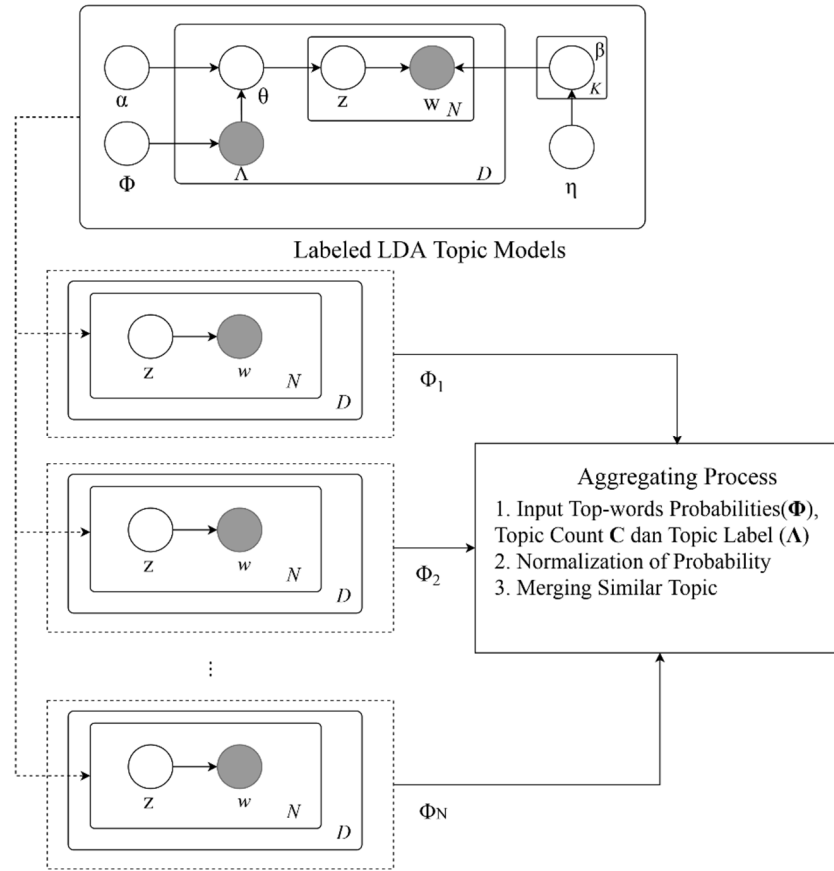


Figure 5 Aggregated Topic Model Plate Notation and Schema

For different topic labels but 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 combining standard graphical model plate notation (Figure 5), extended with an aggregating process notion. We begin the description of ATM by some definitions, following the notation of [68]. Firstly, we define 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 session works on an equally sized partitions of the dataset,

$$\mathbf{T} = \{\Phi_1, \Phi_2, \dots, \Phi_N\} \quad (1)$$

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 7), each having K topic:

$$\Phi_i = \{\varphi_1, \varphi_2, \dots, \varphi_K\} \quad (2)$$

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

$$\varphi_k = p(w|z = k) \quad (3)$$

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

$$p(w|z = k) = \{P_{\varphi_k}(w_1), P_{\varphi_k}(w_2), \dots, P_{\varphi_k}(w_v)\} \quad (4)$$

We can implement φ as a dictionary, each of the entry 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:

$$\begin{aligned} \mathbf{C} &= c(\varphi_1), c(\varphi_2), \dots, c(\varphi_k) \\ \mathbf{\Lambda} &= \lambda(\varphi_1), \lambda(\varphi_2), \dots, \lambda(\varphi_k) \end{aligned} \quad (5)$$

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

$$c(\varphi k) = |\{z \mid z_{m,n} = k\}| \quad (6)$$

Next, we are going to briefly describe the aggregation process to merge several labeled topic models into one. Firstly, the aggregation process needs to use merging function between two topics that has same labels (see Figure 6). The concept of merge is to recalculate the probability of each word component based 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 with all 44,280 unique labels in the main 650K corpus.

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

Figure 6 Aggregate procedure and Merge function to form the aggregated model

This merging function will be invoked from inside the aggregate function, which essentially looks for any two or more topics which have the same label and merges it.

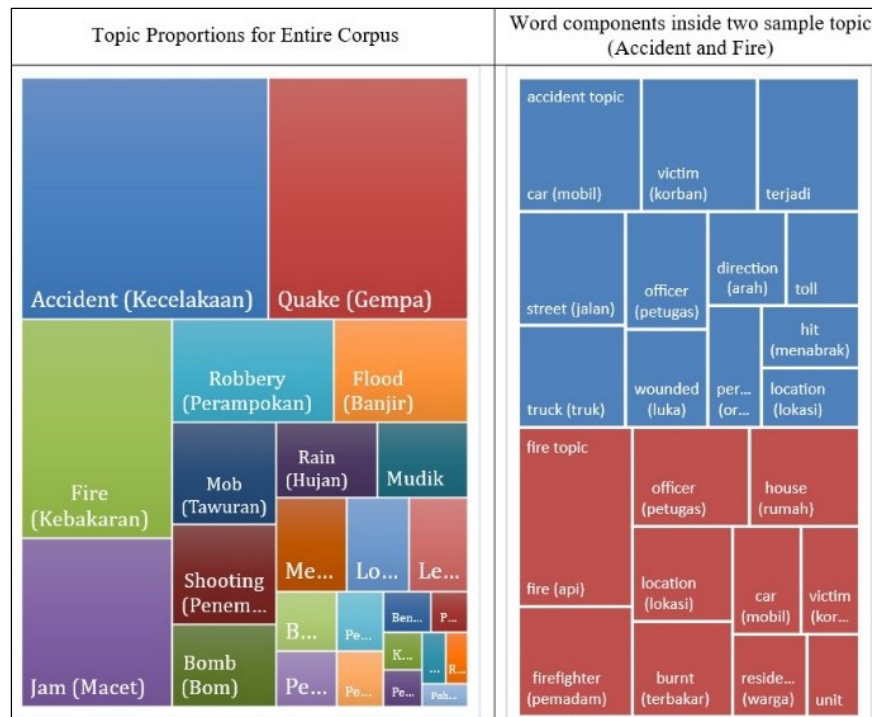


Figure 7 Treemap of Topic Proportions (left) and the top-words from Two Sample Topics (accident and fire) (right). The area shown on the left figure is determined by the number of topic assignments to that particular label / $C(\varphi)$. The area shown on the right figure is determined by the probability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

The number of the assigned topic represented by the area of the square of

Figure 7. Each of the box is a topic (φ), the area is defined by $C(\varphi)$ that is still decomposable by the (semantically related) keywords that is represented by the top-words w_1, w_2, \dots, w_v variable which each has area proportional to the probability of each word within that topic, $P\varphi k(w_i)$. This provides a selection of words that, along with word embedding selection, comprise of our event keywords and regular expression features.

function topic_sim(φ_1, φ_2):

input: two topics to be evaluated

output: 0 – 1 degree of similarity between the topic input.

begin:

for each w **in** φ_1 :

$num = num + P_{\varphi_1}(w) \cdot P_{\varphi_2}(w)$

$den_1 = den_1 + P_{\varphi_1}(w)^2$

end for

for each w **in** φ_2 :

$den_2 = den_2 + P_{\varphi_1}(w)^2$

end for

if $den_1 = 0$ **or** $den_2 = 0$ **then return** 0

else return $\frac{num}{\sqrt{den_1 \cdot den_2}}$

end

Figure 8 Topic Similarity metric adapted from cosine similarity

4.4. Topic Similarity

The aggregated topics will have all unique set of labels (tags) from all document. In order to see find 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. The pseudo-code can be seen in Figure 8. This metric can be used to cluster similar topics and for taxonomy use such as demonstrated in Figure 14.

4.5. Semantic Gazetteer for Event Keywords feature and Numeric Argument Recognition

The large corpus provide 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[33]. 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 forms the semantic gazetteer, which serves as a lookup method or list of terms with regards to various concept (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 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 Event-keywords feature is basically set of lists trigger keywords related to each major event that obtained by selection of either top-words, or most similar word, or bigrams that has most occurrences. The generated list (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 produced by our Aggregated Topic Model. (n-top-words).
2. Semantic similarity produced by Word2Vec [61] *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 list (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”/ <i>gempa</i> : 1. (‘quake / <i>gempa</i> ’, 0.753) 2. (‘shake / <i>guncangan</i> ’, 0.739) 3. (‘tremble / <i>getaran</i> ’, 0.719) 2. (‘hurricane/ <i>topan</i> ’, 0.710) 3. (‘richter’, 0.693) | semantic related of “quake”/ <i>gempa</i> : 1.(‘shake / <i>mengguncang</i> , 0.00177), 2.(‘repeated / <i>susulan</i> ’, 0.00029), 3.(‘strength / <i>kekuatan</i> ’, 0.00013), 4.(‘scattered/ <i>berhamburan</i> ’, 8.917e-06), 5.(‘cracked/ <i>retak</i> ’, 8.916e-06) | bigrams of “quake”/ <i>gempa</i> : 1. (‘earthquake / <i>gempa bumi</i> ’, 3094), 2. (‘quake with magnitude / <i>gempa berkekuatan</i> ’, 1993), 3. (‘repeating quake / <i>gempa susulan</i> ’, 1062), 4. (‘volcanic earthquake / <i>gempa vulkanik</i> ’, 320), 5. (‘shallow quake / <i>gempa dangkal</i> ’, 27) |

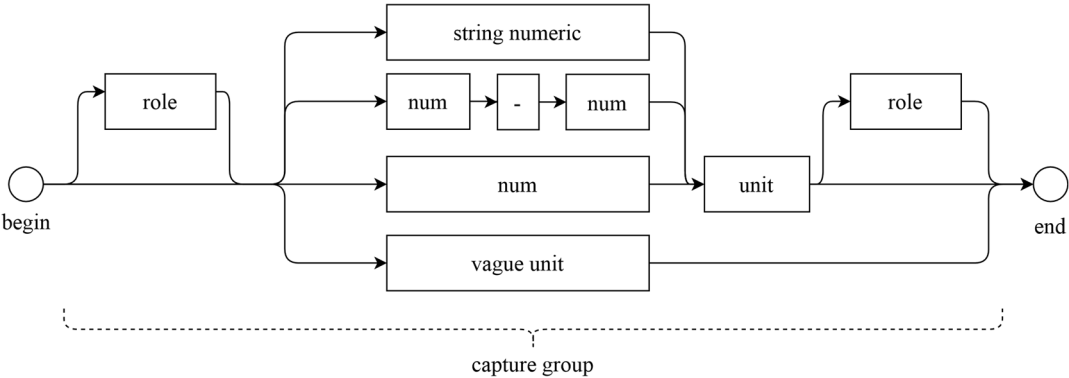
The semantic relatedness and similarity vector obtained from large corpus is also being used to build some regular-expression rule-based feature for entity and numerical argument recognition. This would improve the generalizability of the model, similar to approach [33]. As an example of this feature 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 on Figure 9.

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

Figure 9 Example regular expression for recognizing types of place names. Terms separated by | (or) are composed from semantic similarity from names of river, settlements and mountains respectively.

The next use of concepts keywords within semantic gazetteer is to build 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 [69], 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 10). 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 expression such as (10-20 *cm*), of which the parser will took an average number later on. Also, a *string numeric* expression means that the regex will be able to detect pattern 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 feature to detect which portion of the document that match the argument for particular event. The feature will be used by the sequence labelling framework. The reason is that the statistical sequence labeller will do more generalization and less “brittle” inference.

Figure 10 Regular expression to detect numerical argument. The argument typically either started or ended with the role keyword followed with various numerical quantities, followed with the unit of the argument. For example, “the accident leaves 2 people killed” will be extracted as 2 (numeric) *people* (unit) *killed* (role).



4.6. 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 result from the toponym resolution process, i.e. the smallest administrative level. The motivation assumes that news article will report most precise toponym as 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 will be combined with Event feature from the earlier classified event trigger so that the classifier algorithm can acknowledge the peculiarity for particular event such as Earthquake, which is often occurs or affect several provinces or even countries. The feature is referenced as Smallest Administrative Level (SAL) within document scope that is found by

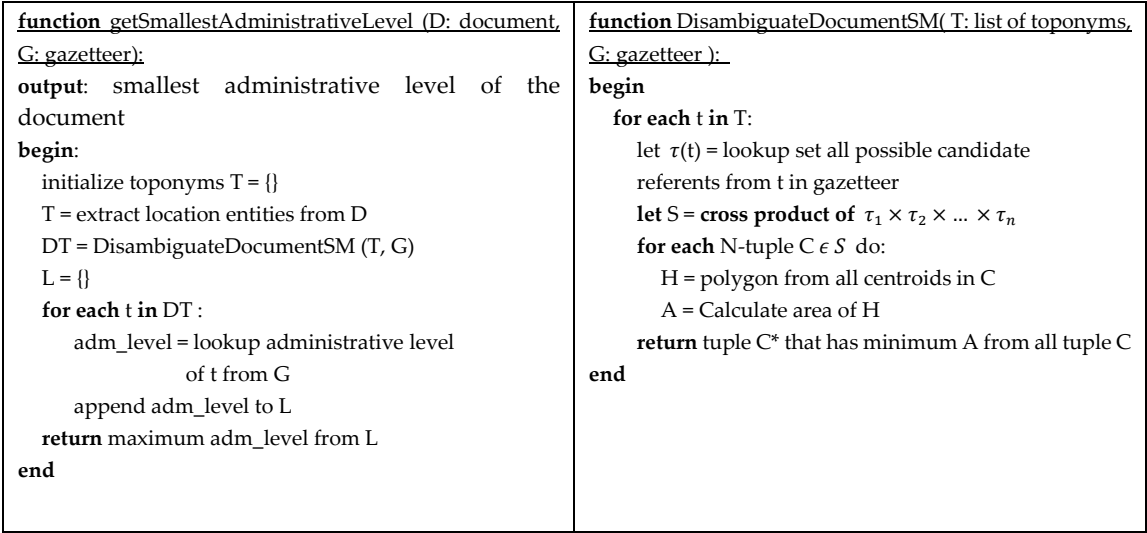


Figure 11 Algorithm for finding Smallest Administrative Level feature using Spatial Minimality (SM) from [12].

disambiguation process for each toponyms found in the document using spatial minimality (SM) (see Figure 11) and spatial minimality centroid distance administrative (SMCD-ADM). 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. This is useful for speeding up the process and to avoid the degenerate cases where there is only two or less toponym inside document. In other word, the minimality of area is replaced by the minimality of the distance of polygon candidates to its centroid.
- 2) The minimality distance is adjusted by multiplying it with the administrative level of an area. Hence, the smaller administrative 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.

```

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
    maxdistC = find point p  $\in$  C that has maximum distance to centroid Cd
    adjusted_maxdistC = (adm_level + 1) · M · maxdistC
  return tuple C that has smallest maxdistC
end

```

Figure 12 Modified Spatial Minimality with Centroid Distance and adjustment factor based on Administrative level.

Note that smallest administrative level corresponds to the maximum integer indicated on administrative level field³ in the case of our chosen gazetteer (GADM). Then, the binary feature is calculated, by simply comparing whether the particular token toponym's administrative level equals to the smallest administrative level or not. The feature make use the output of spatial minimality algorithm to disambiguate document from the detected toponyms. Hence, basically it uses geometric minimality heuristics.

5. Experiments and Results

5.1. Geotagging and Event Extraction

As indicated earlier we approach the geotagging and event extraction as a sequence labeling problem. Geotagging problem in this work is casted 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 classifier from the *sklearn-crfsuite* package. The training was done on 926 sentences, 16,444 tokens on subset of the large corpus with four main topic categories: Quake, Accident, Flood, and Fire. The CRF model is solved using L-BFGS algorithm with 100 maximum iterations and 10-fold cross validation. This makes a 90% rolling portion of the training data are fitted against the rest and the best parameter to be found. We tested the standard definition of precision, recall, and the F1-score.

For the entity extraction, we then compare the model with baseline CRF with gazetteer and POS tag features without including the event-keyword features and regular expression argument extractor. The inclusion of the two features seen a reasonable improvement. 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

| Entity | Baseline CRF with Gazetteer + POS Tag | | | CRF with Gazetteer + POS Tag + Event-Keywords + Regex | | |
|--------|---------------------------------------|-------|-------|---|-------|-------|
| | P | R | F1 | P | R | F1 |
| B-ARG | 0.832 | 0.712 | 0.767 | 0.845 | 0.739 | 0.788 |
| I-ARG | 0.824 | 0.739 | 0.779 | 0.849 | 0.768 | 0.806 |
| B-EVE | 0.850 | 0.800 | 0.824 | 0.852 | 0.817 | 0.834 |
| I-EVE | 0.824 | 0.609 | 0.700 | 0.789 | 0.652 | 0.714 |
| B-LOC | 0.886 | 0.910 | 0.898 | 0.897 | 0.897 | 0.897 |
| I-LOC | 0.898 | 0.964 | 0.930 | 0.930 | 0.964 | 0.946 |

³ the bigger the code number, the smaller region. Currently the largest number is 4 indicating village administrative level.

| | | | | | | |
|---------------------|-------|-------|-------|-------|-------|-------|
| B-ORG | 0.909 | 0.645 | 0.755 | 0.825 | 0.758 | 0.790 |
| I-ORG | 0.792 | 0.576 | 0.667 | 0.769 | 0.758 | 0.763 |
| <i>micro avg</i> | 0.848 | 0.759 | 0.801 | 0.853 | 0.794 | 0.823 |
| <i>macro avg</i> | 0.852 | 0.744 | 0.790 | 0.844 | 0.794 | 0.817 |
| <i>weighted avg</i> | 0.846 | 0.759 | 0.798 | 0.852 | 0.794 | 0.821 |

Event extraction (event trigger classification and event argument extraction) test is done by training CRF again, but with the predicted Entities fetched from the earlier Entity Extraction phase.

Table 7 Event Trigger Classification Performance

| Event | Baseline CRF + Gazetteer + POS Tag features | | | CRF + Event-Keywords + Entity | | |
|---------------------|--|-------|-------|-------------------------------|-------|-------|
| | P | R | F1 | P | R | F1 |
| ACCIDENT-EVENT | 1.000 | 0.868 | 0.930 | 1.000 | 0.974 | 0.987 |
| FIRE-EVENT | 0.882 | 0.938 | 0.909 | 0.970 | 1.000 | 0.985 |
| FLOOD-EVENT | 0.794 | 0.794 | 0.794 | 1.000 | 1.000 | 1.000 |
| QUAKE-EVENT | 0.793 | 0.852 | 0.821 | 0.964 | 1.000 | 0.982 |
| <i>micro avg</i> | 0.869 | 0.863 | 0.866 | 0.985 | 0.992 | 0.989 |
| <i>macro avg</i> | 0.867 | 0.863 | 0.864 | 0.983 | 0.993 | 0.988 |
| <i>weighted avg</i> | 0.875 | 0.863 | 0.867 | 0.985 | 0.992 | 0.989 |

Table 8 Argument Extraction Performance

| Argument | Baseline CRF with Gazetteer + POS Tag features | | | CRF with Event-Keywords + Entity | | |
|---------------------|---|-------|-------|----------------------------------|-------|-------|
| | P | R | F1 | P | R | F1 |
| DeathVictim-Arg | 0.667 | 0.596 | 0.629 | 0.929 | 0.830 | 0.876 |
| Vehicle-Arg | 0.857 | 0.600 | 0.706 | 0.806 | 0.833 | 0.820 |
| Height-Arg | 0.692 | 0.783 | 0.735 | 1.000 | 0.957 | 0.978 |
| OfficerOfficial-Arg | 0.925 | 0.495 | 0.645 | 0.817 | 0.949 | 0.879 |
| Time-Arg | 0.926 | 1.000 | 0.962 | 0.935 | 1.000 | 0.966 |
| Place-Arg | 0.867 | 0.791 | 0.827 | 0.955 | 0.932 | 0.943 |
| Street-Arg | 0.783 | 0.545 | 0.643 | 0.763 | 0.879 | 0.817 |
| Strength-Arg | 0.857 | 0.750 | 0.800 | 1.000 | 0.688 | 0.815 |
| <i>micro avg</i> | 0.847 | 0.735 | 0.787 | 0.908 | 0.919 | 0.914 |
| <i>macro avg</i> | 0.822 | 0.695 | 0.743 | 0.901 | 0.883 | 0.887 |
| <i>weighted avg</i> | 0.851 | 0.735 | 0.779 | 0.914 | 0.919 | 0.914 |

The above results displayed the baseline performance vs highest performance of particular combination of feature. To see which features combinations that contributes most to the performance of the system, we conducted the *ablation test* for each of the four sequence labeling steps within the phases where sequence labeling is applied. There are 9 features in total to be tested, of which some subset of possible feature combinations feature label displayed in the leftmost column (testing and analyzing all 2^9 combination is prohibitive for our resource). The enabled features are represented by blue box, while the disabled feature represented by grey box. The performance of the particular combination is displayed in 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). The entity labels 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. Note that in this ablation test we refrain from doing cross validation to speed up the process.

On the first step, entity extraction, the argument regex (*arg_regex*) feature offers most improvement compared to other features. The *event keywords* added slight improvement too. The inclusion of *arg_regex*, *org_regex*, and *event keywords* added 2.88% improvement over the baseline (*postag* and *gazetteer*).

The next step (skipping the toponym recognition for the next section because we are only discussing sequence labeling task on this section) is the *event trigger classification*, where we found that the *entity* feature from the earlier entity extraction step had the most contribution to the performance. The *event keywords* (code *ev_keywords*) feature added slight improvement only. The inclusion of *entity* and *ev_keywords* together improved 14.07%. If we combine those features with argument and organization detector regex (*arg_regex* and *org_regex*), it will drop the performance instead. This is likely because the target output (event type label) is not directly correlated with the detected arguments.

On the third step, again, the *entity* features greatly helped the argument classification. In this step, adding *postag* and *arg_regex* feature did not increase performance. The *ev_keywords* is also used to add the performance a little, along with removing *gazetteer* and *postag* feature. The best combination is *ev_keywords* + *entity* feature, which is an improvement of 17.32% over the baseline.

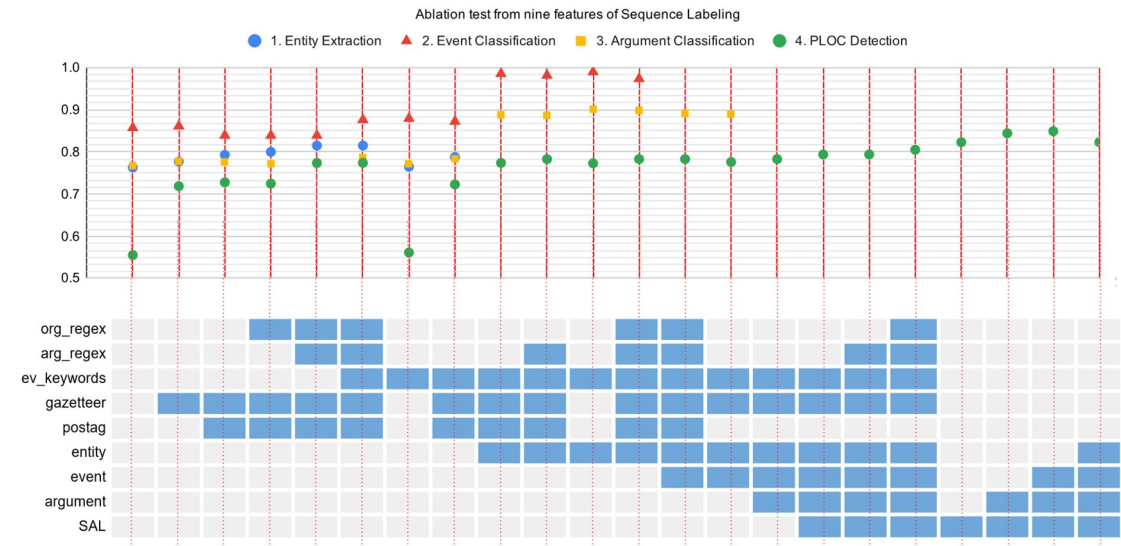


Figure 13 Ablation Test of weighted F1 sore from nine combinations of features of four geotagging steps (step 1, 3, 4, and 5 from Fig. 4). Active features are marked as blue cells (below part of the graphic). Missing score points mean that one or more features is not applicable on that particular step.

5.2. The Pseudo-location Classification

In this fourth step setting, the objective is that every toponym in the corpus is attached a 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 SAL feature is a simple way to boost the F1 score. Adding the *argument* and *event* feature will add the performance more. This shows that event semantics can actually aim in the identification of pseudo-location entities. However, when testing with the 10-fold we found that only pairing argument with SAL (not adding *event* feature) resulted the best performance with improvement of weighted F1 score 19.4% from the baseline. The SAL binary feature is explained in the last part of the Approach Section (Sec. 4).

Table 9 Pseudo-location Detection

| Tag | Baseline + Gaz + Postag | SAL + Argument |
|-----|-------------------------|----------------|
|-----|-------------------------|----------------|

| | P | R | F1 | P | R | F1 |
|---------------------|-------|-------|-------|-------|-------|-------|
| PLOC | 0.826 | 0.667 | 0.738 | 0.832 | 0.912 | 0.870 |
| LOC | 0.675 | 0.651 | 0.663 | 0.867 | 0.756 | 0.807 |
| <i>micro avg</i> | 0.754 | 0.660 | 0.704 | 0.845 | 0.845 | 0.845 |
| <i>macro avg</i> | 0.750 | 0.659 | 0.700 | 0.849 | 0.834 | 0.839 |
| <i>weighted avg</i> | 0.761 | 0.660 | 0.706 | 0.847 | 0.845 | 0.843 |

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 [57] while the latter works often use the topic coherence metric, introduced in [70]. The one used in this experiment, topic coherence, is a metric that measure the quality of the produced topic model given by the co-occurrence of the top words in a particular topic. The more coherence score towards zero, the higher 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:

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

where $D(v)$ is document frequency, i.e. the number of documents that has word v at least once. $D(v_1, v_2)$ is co-document frequency, defined as number of documents which has both word 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 .

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) ⁴ | 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 |

We compared the coherence metric using the following approach:

1. LDA implementation of MALLET (LDA via Gibbs Sampler) [71]
2. LDA implementation of Gensim (LDA Variational Bayes) [72]
3. Labeled LDA (LLDA) code that is implemented inside MALLET [71].
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.

⁴ Merge based on top 3 similarity

To explore some thematic space from the corpus we are interested to obtain 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 at Figure 8. From the seed topic label (for example “jakarta flood” / “banjir jakarta”), we limit to 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 14. Obtaining this result is not doable straightforward from Labeled LDA because the memory limitation on the number of unique document tags.

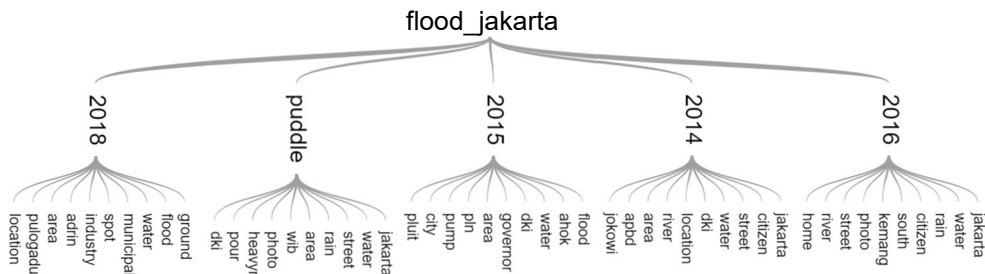


Figure 14 Taxonomy Generated (translated) by Topic Similarity Metric from seed root node (topic tag) “banjir_jakarta” (jakarta_flood). The leaf nodes are the top-words of its respective parent topic. The generated tree is limited to 5 most similar topic.

5.4. Disambiguation and Toponym Resolution

We are testing 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 simply 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.

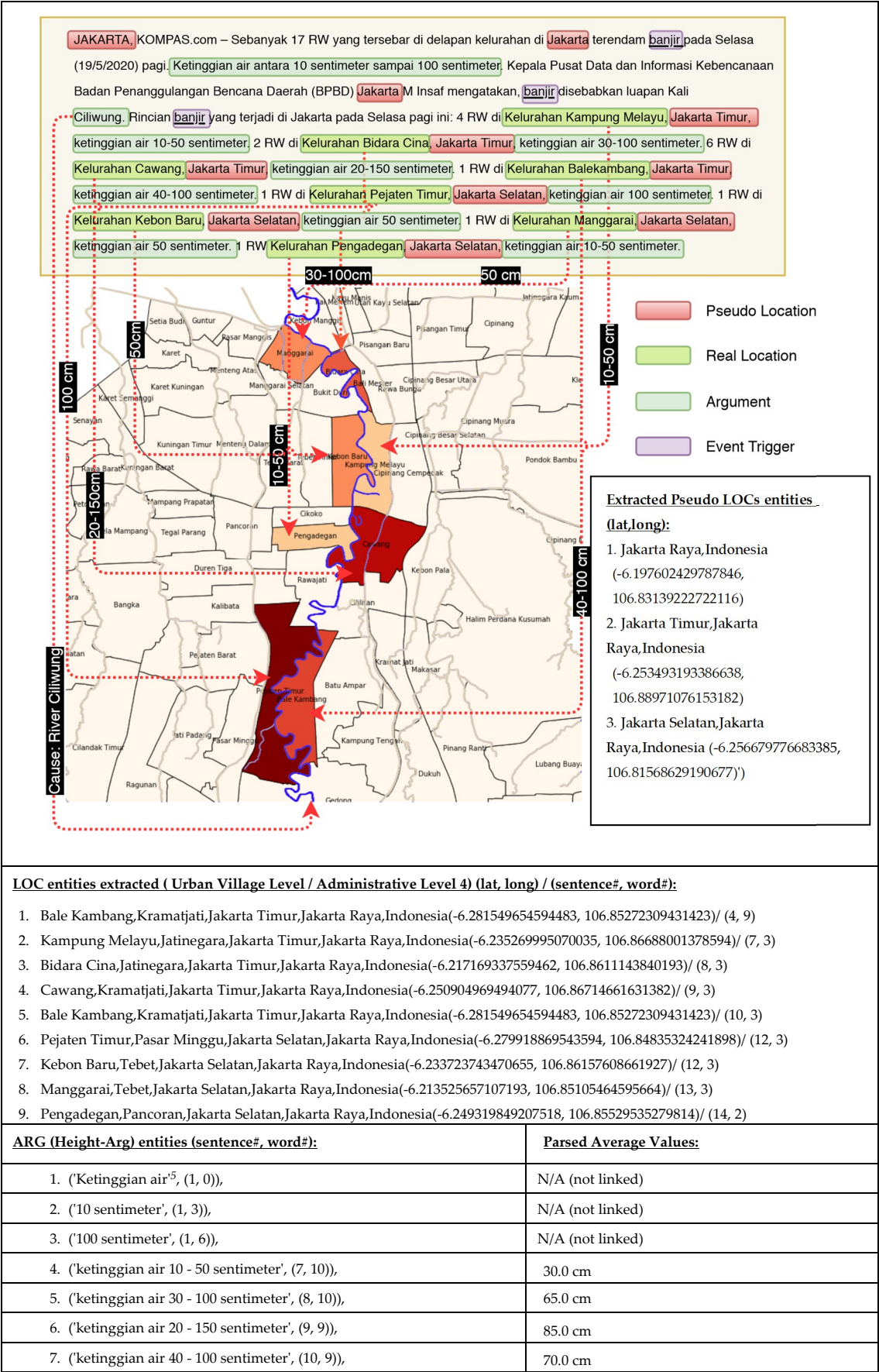
Table 11 Toponym Resolution Performance (Accuracy)

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

5.5. Auto Generation of Thematic Map

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 event trigger class is dominantly about flood and thus led to the typical argument extraction for the flood. 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.

In the case of the article, 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 argument, these numbers will be averaged before it linked to 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 pr ovided 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 below.



⁵ Note: 'Ketinggian air' means the height of the water during the flood. Bold text are the Location extracted (not including pseudo LOCs).

| | |
|--|-------------------------------------|
| 8. ('ketinggian air 100 sentimeter', (11, 10)), | 100.0 cm |
| 9. ('ketinggian air 50 sentimeter', (12, 10)), | 50.0 cm |
| 10.('ketinggian air 50 sentimeter', (13, 9)), | 50.0 cm |
| 11.('ketinggian air 10 - 50 sentimeter', (14, 8)), | 30.0 cm |
| 12.('ketinggian air 10 - 50 sentimeter', (15, 9)) | 30.0 cm |
| Cause-Arg: Kali Ciliwung', (2, 21) | <i>Overlaid as blue line in map</i> |

Figure 15 Visualization sample from a single article with result from our proposed Event Geoparser.

6. Discussion

In the first experiment we are testing the combination of features for the three phases of event geoparsing. The model is equipped with event keywords features and regular expression rules compiled from semantic gazetteer, which improves entity recognition by a margin of 2.88% (weighted F1 on entity extraction), 14.07% (event trigger classification), and argument extraction by 10.74%. The list of keywords derived from semantic gazetteer typically able to improve the recall as it provided related and similar keywords that might not be seen in the development set, thus preventing overfitting that might hinder model generalization. The small improvement margin probably can be explained to the relatively small corpus size. It can also be seen that the Pseudo-location identification task had improved 12.1% in order to discriminate locative and precise toponyms related to an event with pseudo-location entities. This proved that event semantics supplied by the event extraction method is able to improve geoparsing with event-level scope resolution.

The event keywords and regular expression features are of binary type, which is suited within CRF sequence labeling model well. However, a scalar metric might be more desirable for another model. This particular result served as evidence that integration of geoparser method (disambiguation to the correct administrative level and coordinate) with event extraction technique is fruitful and worth to be researched.

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 large number of label (extreme labeling problem) that we often find in web news portal 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.

The event extraction framework used in this work is still using local, per sentence features, except 1) the tags result for each phase 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 [73] and [33] uses global features and joint model to perform the event extraction task, and its integration a worth to pursue. Also worth to mention that the task of event extraction can be structured (due to its similarity) as dependency parsing task, with semantic roles represents the dependent entities to the event anchors or trigger [74].

With the event extraction geoparser put into place, we then have the choropleth map visualized automatically for flood topic (Figure 15). Darker tone means higher the 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. It can be seen the interplay between extracted event semantics and inferred geospatial location provided. This may be used as richer data for generating various thematic maps and further geospatial analysis. Arguably, looking from thematic map is easier and faster in delivering geospatial information across to human reader. The map can be considered as exploratory analysis to augment the geospatial event information presented in text and gives the reader a better understanding of it.

7. Conclusion

Geoparsing and event extraction are both active research topics and has been around for more than a decade. The recent works of geoparser are being more equipped with machine learning and natural language processing methods to better cope with text data from the internet. 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. The work described in this paper described novel approach that integrate geoparsing and provides implementation and evidence on how the integration of geospatial based feature can benefit the geoparsing workflow by improving the Pseudo-location detection as a part of the 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, and 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.

We also tested 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 is able to improve the generalizability of the model. The coherence test shows an acceptable performance of ATM even with very large number of topics (K). Thus, it is a valuable tool for exploring semantic relatedness especially with multi labeled corpora.

Also contributed by this work is the event geoparsing news corpus in Bahasa Indonesia which offered new testbed on extraction of events and event argument along with geoparsing task. This may serve to expedite the research of further event extraction framework. In the future, we plan to develop next pipeline which integrate a visual retrieval system as additional component to the extraction and geoparsing method described here, which serves automatically generated thematic maps from attribute data.

It must be noted that we are only measuring the accuracy and F1 performance. The integrated event extraction as described will add several layers of processing. This may stand as a disadvantage in the terms of runtime of execution of the model, especially in large scale setting such as in GDELT scale. In the future work, we plan to includes the addition of dependency parsing and more sophisticated event coreference resolution. The geoparser can also be employed to a GIR framework in order to provide a richer geographical retrieval system.

References

- [1] M. Himmelstein, "Local Search : The Internet Is the Yellow Pages," *Computer (Long. Beach. Calif.)*, vol. 38, no. 2, pp. 26–34, 2005.
- [2] M. Wunderwald, "NewsX: Event Extraction from News Articles," 2011.
- [3] J. Gelernter and S. Balaji, "An algorithm for local geoparsing of microtext," *Geoinformatica*, vol. 17, no. 4, pp. 635–667, Jan. 2013, doi: 10.1007/s10707-012-0173-8.
- [4] W. Wang and K. Stewart, "Spatiotemporal and semantic information extraction from Web news reports about natural hazards," *Comput. Environ. Urban Syst.*, vol. 50, pp. 30–40, 2015, doi: 10.1016/j.compenvurbsys.2014.11.001.
- [5] C. C. Freifeld, K. D. Mandl, B. Y. Reis, and J. S. Brownstein, "HealthMap : Global Infectious Disease Monitoring through," *J. Am. Med. Informatics Assoc.*, vol. 15, no. 2, pp. 150–157, 2008, doi: 10.1197/jamia.M2544.Introduction.
- [6] R. S. Purves, P. Clough, and C. B. Jones, "The Design and Implementation of SPIRIT : a Spatially- Aware Search Engine for Information Retrieval on the Internet," *Int. J. Geogr. Inf. Sci.*, pp. 717–745, 2007.
- [7] M. Gritta, M. T. Pilehvar, and N. Collier, "A pragmatic guide to geoparsing evaluation," *Lang. Resour. Eval.*, 2019, doi: 10.1007/s10579-019-09475-3.
- [8] A. G. Woodruff, "(GIPSY) Georeferenced Information Processing System," *J. Am. Soc. Inf. Sci.*, vol. 45,

- no. 9, pp. 1–44, 1994.
- [9] M. Gritta, “Where are you talking about ? Advances and Challenges of Geographic Analysis of Text with Application to Disease Monitoring,” 2019.
 - [10] A. Bo, S. Peng, T. Xinming, and N. Alimu, “Spatio-temporal visualization system of news events based on GIS,” *2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011*, pp. 448–451, 2011, doi: 10.1109/ICCSN.2011.6014089.
 - [11] C. Grover *et al.*, “Use of the Edinburgh geoparser for georeferencing digitized historical collections,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 368, no. 1925, pp. 3875–3889, 2010, doi: 10.1098/rsta.2010.0149.
 - [12] J. L. Leidner, “Toponym resolution in text: Annotation, evaluation and applications of spatial grounding of place names,” The University of Edinburgh, 2008.
 - [13] E. Amitay, N. Har’El, R. Sivan, and A. Soffer, “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] M. Karimzadeh, S. Pezanowski, A. M. MacEachren, and J. O. Wallgrün, “GeoTxt: A scalable geoparsing system for unstructured text geolocation,” *Trans. GIS*, vol. 23, no. 1, pp. 118–136, 2019, doi: 10.1111/tgis.12510.
 - [15] M. Gritta, M. T. Pilehvar, and N. Collier, “Which Melbourne? Augmenting geocoding with maps,” *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 1, no. 2, pp. 1285–1296, 2018, doi: 10.18653/v1/p18-1119.
 - [16] C. D’Ignazio, R. Bhargava, E. Zuckerman, and L. Beck, “CLIFF-CLAVIN: Determining Geographic Focus for News Articles,” *Proc. NewsKDD Data Sci. News Publ.*, 2014.
 - [17] M. D. Lieberman, J. Sperling, and D. C. Washington, “STEWART: 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, [Online]. Available: <https://www.ldc.upenn.edu/collaborations/past-projects/ace>.
 - [19] A. Dewandaru, S. I. Supriana, and S. Akbar, “Event-Oriented Map Extraction From Web News Portal : Binary Map Case Study on Diphteria Outbreak and Flood in Jakarta,” *ICAICTA 2018 - 5th Int. Conf. Adv. Informatics Concepts Theory Appl.*, no. November, pp. 72–77, 2018, doi: 10.1109/ICAICTA.2018.8541345.
 - [20] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled LDA : A supervised topic model for credit attribution in multi-labeled corpora,” *Conf. Empir. Methods Nat. Lang. Process.*, no. August, pp. 248–256, 2009, doi: 10.3115/1699510.1699543.
 - [21] B. Technologies, “CLAVIN.” [Online]. Available: <https://github.com/Novetta/CLAVIN>.
 - [22] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling, “NewsStand,” *Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst. - GIS ’08*, vol. 2008, no. November, p. 1, 2008, doi: 10.1145/1463434.1463458.
 - [23] G. Andogah, G. Bouma, and J. Nerbonne, “Every document has a geographical scope,” *Data Knowl. Eng.*, vol. 81–82, pp. 1–20, 2012, doi: 10.1016/j.datak.2012.07.002.
 - [24] H. Li, R. K. Srihari, C. Niu, and W. Li, “Location Normalization for Information Extraction *,” 1996.
 - [25] R. K. SRIHARI, W. LI, T. CORNELL, and C. NIU, “InfoXtract: A customizable intermediate level information extraction engine,” *Nat. Lang. Eng.*, vol. 14, no. 01, pp. 33–69, 2006, doi: 10.1017/S1351324906004116.

- [26] P. A. Schrodtt and K. Leetaru, "GDELT: Global Data on Events, Location and Tone, 1979-2012," *Int. Stud. Assoc. Meet.*, pp. 1–49, 2013.
- [27] K. H. Leetaru, "Fulltext geocoding versus spatial metadata for large text archives: Towards a geographically enriched wikipedia," *D-Lib Mag.*, vol. 18, no. 9–10, pp. 1–23, 2012, doi: 10.1045/september2012-leetaru.
- [28] S. J. Lee, H. Liu, and M. D. Ward, "Lost in space: Geolocation in event data," *Polit. Sci. Res. Methods*, vol. 7, no. 4, pp. 871–888, 2019, doi: 10.1017/psrm.2018.23.
- [29] A. Halterman, "Massachusetts Institute of Technology Political Science Department Linking Events and Locations in Political Text Andrew Halterman , Massachusetts Institute of Technology," 2018.
- [30] M. B. Imani, S. Chandra, S. Ma, L. Khan, and B. Thuraisingham, "Focus Location Extraction from Political News Reports with Bias Correction Focus Location Extraction from Political News Reports with Bias Correction," no. December, 2017, doi: 10.1109/BigData.2017.8258141.
- [31] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1532–1543, 2014, doi: 10.3115/v1/D14-1162.
- [32] A. Halterman, "Geolocating Political Events in Text," pp. 29–39, 2019, doi: 10.18653/v1/w19-2104.
- [33] B. Yang and T. Mitchell, "Joint extraction of events and entities within a document context," in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016, pp. 289–299, doi: 10.18653/v1/n16-1033.
- [34] Jochen L. Leidner and M. D. Lieberman, "Detecting Geographical References in the Form of Place Names and Associated Spatial Natural Language," *SIGSPATIAL Spec.*, vol. 3, no. 2, pp. 5–11, 2011.
- [35] Geonames.org, "Geonames," 2020. <https://geonames.org> (accessed May 31, 2020).
- [36] E. G. Morton-Owens, "A tool for extracting and indexing spatio-temporal information from biographical articles in Wikipedia," 2012, [Online]. Available: http://www.cs.nyu.edu/web/Research/MsTheses/owens_emily.pdf.
- [37] F. Schilder, Y. Versley, and C. Habel, "Extracting spatial information: grounding, classifying and linking spatial expressions," *Proc. Work. Geogr. Inf. Retr. SIGIR 2004*, pp. 1–3, 2004, [Online]. Available: http://publikationen.stub.uni-frankfurt.de/frontdoor/deliver/index/docId/9959/file/VERSLEY_Extracting_spatial_information.pdf.
- [38] R. Lan, M. D. Adelfio, and H. Samet, "Spatio-temporal disease tracking using news articles," *Proc. Third ACM SIGSPATIAL Int. Work. Use GIS Public Heal. - Heal. '14*, no. c, pp. 31–38, 2014, doi: 10.1145/2676629.2676637.
- [39] B. R. Monteiro, C. A. Davis, and F. Fonseca, "A survey on the geographic scope of textual documents," *Comput. Geosci.*, vol. 96, pp. 23–34, 2016, doi: 10.1016/j.cageo.2016.07.017.
- [40] I. Bensalem and M. K. Kholadi, "Toponym disambiguation by arborescent relationships," *J. Comput. Sci.*, vol. 6, no. 6, pp. 653–659, 2010, doi: 10.3844/jcssp.2010.653.659.
- [41] K. Markert and M. Nissim, "Towards a corpus annotated for metonymies: The case of location names," *Proc. 3rd Int. Conf. Lang. Resour. Eval. Lr. 2002*, pp. 1385–1392, 2002.
- [42] F. Hogenboom, "An Overview of Event Extraction from Text," 2011.
- [43] J. Pustejovsky *et al.*, "The Specification Language TimeML," pp. 1–15, 2004.
- [44] W. Wang, D. Zhao, and D. Wang, "Chinese news event 5W1H elements extraction using semantic role labeling," *Proc. - 3rd Int. Symp. Inf. Process. ISIP 2010*, pp. 484–489, 2010, doi: 10.1109/ISIP.2010.112.
- [45] M. L. Khodra, "Event Extraction on Indonesian News Article Using Multiclass Categorization," pp. 1–5, 2015.

- [46] E. Rauch, M. Bukatin, K. Baker, and M. Avenue, "A confidence-based framework for disambiguating geographic terms," 2003.
- [47] J. L. Leidner, G. Sinclair, and B. Webber, "Grounding spatial named entities for information extraction and question answering," 2003.
- [48] M. B. Habib and M. Van Keulen, "A hybrid approach for robust multilingual toponym extraction and disambiguation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7912 LNCS, pp. 1–15, 2013, doi: 10.1007/978-3-642-38634-3_1.
- [49] M. Nissim, C. Matheson, and J. Reid, "Recognizing Geographical Entities in Scottish Historical Documents," *Proc. Work. Geogr. Inf. Retr. SIGIR 2004*, 2004.
- [50] B. Adams, G. Mckenzie, and M. Gahegan, "Frankenplace : Interactive Thematic Mapping for Ad Hoc Exploratory Search Categories and Subject Descriptors," *Proc. 24th Int. World Wide Web Conf. (WWW 2015)*, pp. 12–22, 2015, doi: 10.1145/2736277.2741137.
- [51] D. Buscaldi, "Toponym Disambiguation in Information Retrieval," 2010.
- [52] D. A. Smith and G. Crane, "Disambiguating geographic names in a historical digital library," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2163, pp. 127–136, 2001, doi: 10.1007/3-540-44796-2_12.
- [53] W. Wang, "Automated spatiotemporal and semantic information extraction for hazards," *Thesis*, 2014.
- [54] J. Wang *et al.*, "Biomedical event trigger detection by dependency-based word embedding," *BMC Med. Genomics*, vol. 9, no. Suppl 2, 2016, doi: 10.1186/s12920-016-0203-8.
- [55] D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 55–65, 2010, doi: 10.1109/MSP.2010.938079.
- [56] R. Řehůřek, "Scalability of Semantic Analysis in Natural Language Processing," p. 147, 2011, [Online]. Available: http://radimrehurek.com/phd_rehurek.pdf.
- [57] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [58] Y. Papanikolaou and G. Tsoumakas, "Subset Labeled LDA for Large-Scale Multi-Label Classification," 2017, [Online]. Available: <http://arxiv.org/abs/1709.05480>.
- [59] D. Kang, Y. Park, and S. N. Chari, "Hetero-labeled LDA: A partially supervised topic model with heterogeneous labels," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8724 LNAI, no. PART 1, pp. 640–655, 2014, doi: 10.1007/978-3-662-44848-9_41.
- [60] D. Greene, D. O'Callaghan, and P. Cunningham, "How many topics? Stability analysis for topic models," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8724 LNAI, no. PART 1, pp. 498–513, 2014, doi: 10.1007/978-3-662-44848-9_32.
- [61] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, pp. 1–12, 2013, doi: 10.1162/1532444303322533223.
- [62] J. L. Leidner, "An evaluation dataset for the toponym resolution task," *Comput. Environ. Urban Syst.*, vol. 30, no. 4, pp. 400–417, 2006.
- [63] M. Gritta, M. T. Pilehvar, N. Limsopatham, and N. Collier, "What's missing in geographical parsing?," *Lang. Resour. Eval.*, vol. 52, no. 2, pp. 603–623, 2018, doi: 10.1007/s10579-017-9385-8.
- [64] A. Dewandaru, "EVENT GEOPARSING INDONESIAN NEWS DATASET," *IEEE Dataport*, 2020. .
- [65] E. M. Bender and A. Lascarides, "Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from Semantics and Pragmatics," *Synth. Lect. Hum. Lang. Technol.*, vol. 12, no. 3, pp. 1–268, 2019.
- [66] G. A. Areas, "GADM database of Global Administrative Areas, version 2.0," *Berkeley, CA Univ. Berkeley*,

- 2012.
- [67] A. Purwarianti, A. Andhika, A. F. Wicaksono, I. Afif, and F. Ferdian, "InaNLP: Indonesia natural language processing toolkit, case study: Complaint tweet classification," *4th IGNITE Conf. 2016 Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2016*, pp. 1–5, 2016, doi: 10.1109/ICAICTA.2016.7803103.
 - [68] G. Heinrich, "Parameter Estimation for Text Analysis," *Web http://www.arbylon.net/publications/text-est.pdf*, pp. 1–31, 2008, doi: 10.2514/2.3375.
 - [69] M. A. Murtaugh, B. Smith, D. Redd, and Q. Zeng-treidler, "Regular expression-based learning to extract bodyweight values from clinical notes," *J. Biomed. Inform.*, vol. 54, pp. 186–190, 2015, doi: 10.1016/j.jbi.2015.02.009.
 - [70] 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, [Online]. Available: <http://mimno.infosci.cornell.edu/papers/mimno-semantic-emnlp.pdf>.
 - [71] D. Mimno, "Package 'mallet,'" *Compr. R Arch. Netw.*, pp. 1–11, 2015, [Online]. Available: <https://cran.r-project.org/web/packages/mallet/mallet.pdf>.
 - [72] R. Řehůřek and Petr Sojka, "Software Framework for Topic Modelling with Large Corpora." ELRA, p. 45, 2010.
 - [73] Q. Li, H. Ji, and L. Huang, "Joint Event Extraction via Structured Prediction with Global Features," 2013.
 - [74] 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.
 - [75] R. S. Purves *et al.*, "The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet," *Int. J. Geogr. Inf. Sci.*, vol. 21, no. 7, pp. 717–745, Aug. 2007, doi: 10.1080/13658810601169840.