Rule-based geographical relationship extraction in Dutch news articles

Antoon W. Meijer
11016914

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
dr. M.J. Marx

Innovation Center for Artificial Intelligence(?)
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

January 31st, 2019

Abstract

Contents

1 Introduction

As the amount of published articles and other Dutch text on the Internet continues to increase, it would be helpful for researchers and others interested in location-specific articles to be able to search for articles based on locations mentioned in the article. It would enable them to spend less time sifting through irrelevant articles, and to spend more time doing their own research. A geolocating model could also be beneficial for other purposes, such as automatic traffic accident location statistics.

Olieman et al [1] have attempted to accommodate this need by creating an application, LocLinkVis, that plots Dutch input text on a map based on geographical entities in the text. By doing so, users can more quickly see if an article is relevant. The approach taken by Olieman et al [1] identifies toponyms in text by using a gazetteer populated with OpenStreetMap data. After toponym disambiguation, the most likely candidate is shown on a map. However, LocLinkVis does not take into account that some locations might consist of more than one toponym. In such a case, the application simply shows the location of all involved toponyms, instead of understanding how these toponyms are linked and need to be interpreted as one location.

1.1 Research question

This research is aimed at improving the accuracy of geolocating complex locations mentioned in Dutch articles, through the use of toponyms and surrounding spatial identifier words. Accurately geolocating complex locations is especially important, as non-complex locations are already handled with sufficient accuracy by commonly-used geographical search engines such as OpenStreetMap and Google Maps. In this research, complex locations are defined as locations that involve multiple toponyms to accurately describe. Below is an example of a complex location:

Een 25-jarige man uit Arnemuiden is dinsdagmiddag om het leven gekomen bij een ongeval op de N59 ter hoogte van Oude-Tonge.

In the example above, the location is not the entire length of the N59 road, but only the section near Oude-Tonge. Such locations are usually not accurately geolocated by the common geographical search engines. This research thus aims to answer the following question:

In order to accurately geolocate a complex location, how can the relationship between several toponyms extracted from Dutch text best be modeled?

Using spatial identifier words to model the relationship between toponyms, how accurately can complex locations be geolocated?

1.2 Overview

[NOG SCHRIJVEN]

2 Related work

3 Method

This section outlines ... [INTRO SCHRIJVEN, SAMENVATTING VAN METHOD SECTIE]

3.1 Dataset

Even though the scope of the research is wider, the dataset that was used solely consists of traffic accident news articles. This choice was made because of the fact that traffic accident articles almost always contain toponyms to describe where the accident took place, with a relatively high degree of complex locations as well, thus making it interesting for this research.

The dataset used originates from the website *flitsservice.nl* and has been edited by Hendriks[2] to separate each article's body from its title. The titles of the articles will not be used in this research, because these never contain complex locations due to their short length, and title toponyms are usually repeated in the article. The dataset contains 7784 articles. For testing purposes, the dataset has been split in two parts. The first 4998 articles are used as training set, the last 2786 are used as testing set. The publishing dates of the articles vary between 2003 and 2019.

3.2 Toponym detection

The first step to geolocating an article is to find the toponyms in the text. This was done using the model constructed by Overdijk[3], which is based on LocLinkVis by Olieman et al.[1]. Using the named entity recognition features of spaCy was also considered, but because of superior accuracy results reported by Overdijk[3], his model was chosen.

The model first tokenizes the input text using spaCy, a natural language processing toolkit available for Dutch language processing. Then, it detects municipalities mentioned in the tokenized text and searches for streets that are within the scope of the detected municipalities. There are some drawbacks to this method, such as when there is no municipality detected, other toponyms will also not be detected. This poses a problem for articles that contain locations from outside of the Netherlands. Also, the model does not recognize landmarks and other high-detail geographical features. However, the model does detect highways and secondary roads (N-roads) that are not tied to a specific municipality. The model also classifies the toponyms into several categories, but these labels are not used in this research.

3.3 Data selection

After toponym detection, all sentences that contain no toponyms are deleted. In the sentences that do contain a toponym, all words further than n words away from the closest toponym are also deleted. The value of n will be decided later in section Model Tuning. A toponym with its surrounding words will from now on be referred to as a *span*. In a span, there can be more than one toponym

if there was more than one toponym in its source sentence. [WINDOW SIZE NOEMEN]

3.4 Spatial identifier selection

When dealing with complex locations, the key to accurate geolocation lies in understanding the relation between the toponyms specifying the location. This relation is expressed in spatial identifier words surrounding the toponyms. To obtain a better understanding of what words are important for spatial identification, a part of the training set was manually classified as either containing complex locations or containing non-complex locations. Subsequently, an analysis was done to visualize the difference in word frequencies between all articles and complex-marked articles.

The complex/non-complex annotated dataset contains the toponym spans of the first two hundred articles of the training set. Eighty-two articles are marked as containing complex locations, the rest are marked non-complex. Because spatial identifier words are never behind a toponym, the words located after the last toponym in a span are not counted in this analysis. The word count in both categories is normalized by the amount of words in each respective category.

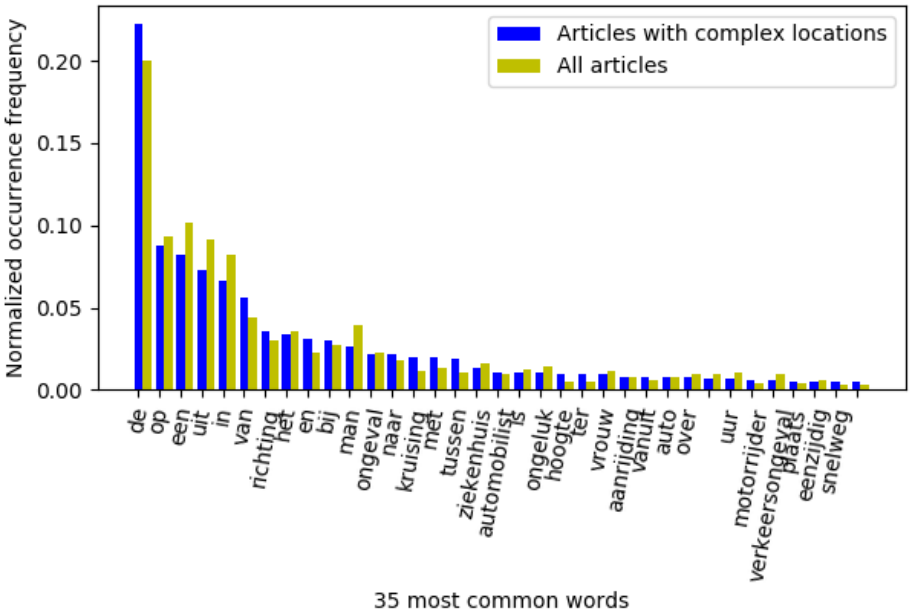

Figure 1: Frequency of thirty-five most common words in complex location articles, together with frequency of said word in all of first two hundred articles.

To get a sense of how accurately this subset of two hundred articles represents the entire training set, Figure 2 was plotted. As one can see, the occurrence frequencies of the most common thirty-five words in both datasets correspond quite well.

Figure 1 shows what words are most frequently occurring in articles containing

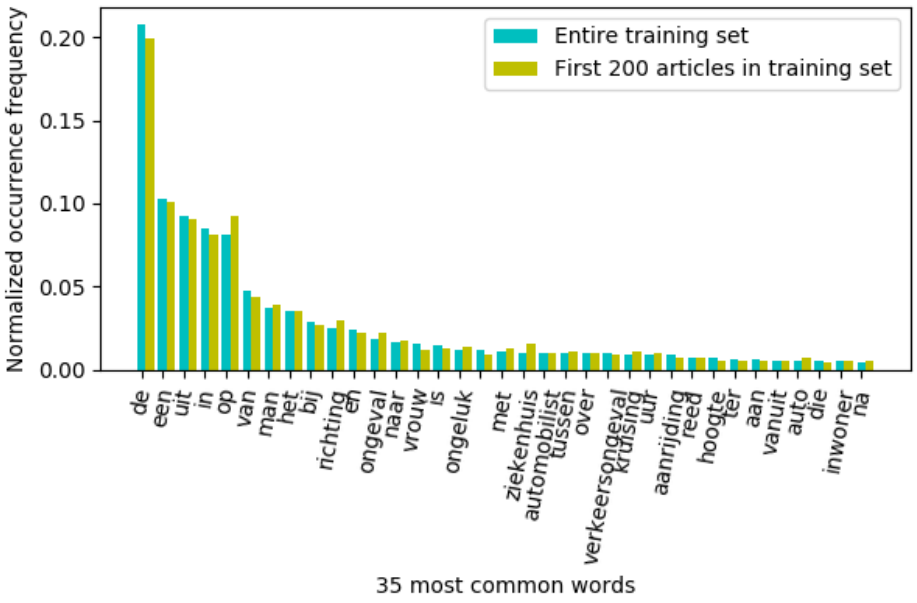

Figure 2: Frequency of thirty-five most common words in the whole training set, together with frequency of said word in first two hundred articles.

complex locations, but it also indicates how often these words are actually occurring in the general training set.

Drawing from Figure 1, word occurrence frequencies are mostly the same between complex locations and average articles, with some minor exceptions. When only looking at words that contain geographical information, there are some words with bigger frequency differences, such as “kruising”,“tussen” and “hoogte”. Even though these words do not appear in the training set very often, they appear to be important to complex locations. Of course, there are also spatial identifiers occurring with similar frequency in both categories that are still very important to understand, such as “op”, “uit”, “in”, “richting”, “bij” and “naar”.

3.5 Parsing spatial identifiers to predicates

Now that the most common spatial identifiers are selected, they can be parsed into specific rules to make geocoding possible. In the table below, the resulting predicates can be found on the left-hand side, with the selected input words on the right-hand side. The number behind the predicates determines how many locations the predicate takes. Some input words, such as “kruispunt” and “te”, are not among the most common thirty-five words, but can easily be added to an existing predicate since they have the same meaning. Sometimes, there occurs more than one of the selected prepositions within the window size before the toponym. In this case, the preposition closest to the toponym is selected.

Predicate	Spatial identifier
ON(1)	op over
IN(1)	in te
AT(1)	bij hoogte
BETWEEN(2)	tussen
INTERSECT(1/2)	kruising kruispunt splitsing

3.6 Converting toponym-predicate pairs into coordinates

After converting spatial identifier words to predicates, each toponym is converted to coordinates in a way that depends on the predicate. This step gives two types of results: a list of coordinate points from road toponyms, processed by "ON(1)" or "INTERSECT(1/2)", or a simple two- or four-coordinate bounding box from other toponyms, processed by the other predicates. Both resulting types have their own result list where they are added to. These lists are later combined, this process is described in section 3.7.

3.6.1 Coordinate source & search method

To retrieve the coordinates associated with a toponym, data is extracted from OpenStreetMap. This is done using Nominatim and Overpass API. Nominatim is used to search for cities and villages with a clear center and surrounding bounding box, whereas the Overpass API is used to search for streets. Nominatim is the preferred search method due to its speed and ease-of-use, but it shows poor recall of highway segments and other long road segments. Both search methods are programmed to primarily search for coordinates in the Netherlands, with Overpass excluding any results abroad, and Nominatim preferring results in the Netherlands.

3.6.2 ON(1)

Through observation, the toponym following the "ON" predicate has been determined to mostly be a street of some kind. Therefore, the toponym coordinates are searched for through Overpass. No bounding box is given other than the Netherlands, so search results usually include multiple streets throughout the country if the street name happens to be a commonly used one. The toponym and result coordinates are added to the list of road search results.

3.6.3 IN(1)

Toponyms following "IN" are largely cities, villages, landmarks and public buildings. Toponyms are thus searched for using Nominatim. Search results of the class "boundary" or "place" are preferred, because both results contain accurate bounding box of a city or village. If these classes are not found, the first search result is returned. The resulting coordinates are added to the list containing all found bounding boxes in that specific article.

3.6.4 AT(1)

3.6.5 BETWEEN(2)

Toponyms occurring after "BETWEEN" can be of any kind, though most are cities and villages. Both toponyms are looked up through Nominatim, with the result of class "place" given priority. In case there is no result of class "place", the first result is returned.

The bounding box between these two toponyms takes the shape of a square, where both toponyms are at opposite ends of a diagonal in the square. The other two points forming the bounding box are located at an angle of forty-five degrees from both sides of the diagonal, according to the properties of a square. To determine the distance between either point and either toponym, the length of the diagonal is divided by the square root of two.

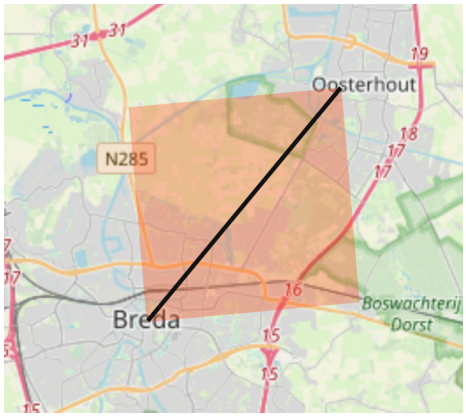

Figure 3: Example of bounding box created by "BETWEEN" predicate. Red zone is bounding box, black line is diagonal between toponyms.

3.6.6 INTERSECT(1/2)

Toponyms occurring in this predicate are assumed to be streets. This predicate sometimes occurs with only one toponym. In this case, the other toponym that makes up the intersection is usually located somewhere else in the sentence or in a neighbouring sentence. Therefore, "INTERSECT(1)" is not processed, but immediately added to the coordinate point result list. This predicate will be dealt with later on, when the two resulting lists are merged.

In the case of "INTERSECT(2)", both toponyms are looked up through the Overpass API. The Overpass API searches for nodes where both toponyms overlap and it returns the coordinates of all nodes where this occurs. The resulting coordinates are added to the list of road search results.

3.7 Predicate result merging

Because street toponym predicates retrieve coordinates for all streets in the Netherlands with the toponym name, these need to be filtered in order to be of any use. That is why "INTERSECT(2)" and "ON(1)" predicate results are matched against any bounding box in the bounding box result list. If any of the result point coordinates fall within a bounding box, all coordinates that lie outside of the bounding box are removed from the coordinate point list, and the bounding box itself is also removed. This is how street coordinate results are narrowed down.

The still unprocessed "INTERSECT(1)" predicate-toponym pair is matched using "INTERSECT(2)", together with another toponym coming from other "INTERSECT(1)" or "ON(1)" predicates that are in the street result list. If any matching nodes are returned, both input toponyms and their predicates are deleted from the street result list.

Any entries in both result lists that do not match with any other entry remain in the list. The merging method described above is used after processing all predicate-toponym pairs in a sentence. Afterwards, all remaining entries in both lists are added to two similar lists that contain locations from the entire article. When all sentences in the article are processed, the merging process is run again these two lists. The reason for performing both sentence-based and article-based merging instead of only article-based merging, is that toponyms in the same sentence are more likely to be related than toponyms in the article.

3.8 Resulting output

The model outputs two lists per article, with one list containing bounding boxes, and the other list containing point coordinates to signify streets and other points. both

4 Results

4.1 Evaluation method

To evaluate the model described in section 3, the first eighty articles of the test set were annotated. Every toponym in the article was manually tagged, including the spatial identifier in front of each toponym. Also, every location formed by the toponyms was annotated. Each location was marked by its coordinates, the toponym that describes the location most specifically, and a precision radius in meters. All locations consisting of toponyms with spatial identifiers outside the set selected for this research were ignored, since the model does not use these toponyms. The eighty articles contain a total of eighty-seven annotated locations. To isolate the model from imperfect recall during toponym detection, the model will also be evaluated using the annotated toponyms with spatial identifiers as input.

To calculate the accuracy the model, the number of correct geolocations is divided by the number of total geolocations. To evaluate a performed geolocation by the model, the average coordinate is calculated for each bounding box or road coordinate list. If this average coordinate is within radius distance of the annotated coordinate, the geolocation is marked correct. If it is outside radius distance, the geolocation is marked incorrect.

	Accuracy
Full model	0.33
Annotated toponym input	0.53

5 Discussion

- Can this study be generalized to all Dutch text, although the dataset contains only traffic accident articles? - Some roads and other toponyms may have changed since the publication date of the article. - Center of a bounding box of a village is sometimes far away of actual center of village. Possibly outside of radius range.

6 Conclusion

References

[1] A. Olieman, J. Kamps, and R. M. Claros, "Loclinkvis: A geographic information retrieval-based system for large-scale exploratory search," *arXiv preprint arXiv:1509.02010*, 2015.

[2] B. Hendriks, "Retrieving, cleaning and analysing dutch news articles about traffic accidents," 2019.

[3] C. Overdijk, "Location recognition and geocoding in dutch traffic news articles," 2020.