Which Portland Is It? A Machine Learning Approach

Nicole R. Schneider
University of Maryland
College Park, MD
nsch@umd.edu

Hanan Samet
University of Maryland
College Park, MD
hjs@umd.edu

ABSTRACT

This paper reviews several approaches to the problem of toponym resolution for news articles referring to 'Portland.' We train several models to differentiate between Portland, Maine and Portland, Oregon, generating features using only the text of the articles. The data used is in the form of articles pulled from NewsStand. The labels, which are provided by NewsStand's interpretation of the articles, allow for a supervised learning approach. We apply Natural Language Processing (NLP) and data cleaning techniques to process the article data, perform feature reduction, and then feed the data to the models. We show that the logistic regression model performs the best of the four models that we test. We also demonstrate that this model learns a more robust representation of the two classes than the other three models do.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**.

KEYWORDS

Toponym resolution, machine learning, geotagging, experimental, spatio-textual

ACM Reference Format:
Nicole R. Schneider and Hanan Samet. 2021. Which Portland Is It? A Machine Learning Approach. In *5th ACM SIGSPATIAL International Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising (LocalRec'21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3486183.3491066

1 INTRODUCTION

With the widespread use of internet-enabled mobile devices, there is a high demand for relevant location-based news articles and information. The news outlets that publish this information tend to cater to a particular geographic domain, based on the location of their readership. This means that news articles are often written with a particular geographic scope in mind, and the intended audience is typically aware of the general spatial region included in that scope. For instance, a news article written by a local newspaper in Portland, Oregon about an event that occurred in that city, may expect that the readers are from Oregon or the surrounding area, and will understand implicitly that a reference to 'Portland,' indicates Portland, Oregon and not any of the other Portlands that exist in the world. This reliance on implicit understanding between writer and reader can be problematic in a world where news articles can easily be searched online and read by audiences more widespread than a writer may have had in mind when publishing the article. This disconnect in scope becomes particularly problematic when articles refer to cities or *toponyms* with common names, including Portland. Without the proper context, these toponyms are ambiguous, and a correct geographic interpretation of the toponym is illusive. Generally speaking, the problem of identifying which city an ambiguous reference is actually referring to is known as toponym resolution, and it is an open area of research.

The problem of toponym resolution has been approached in various ways, including using outside information from Wikepedia [20, 21] and other web pages [8], and exploiting metadata and geotag information [9]. However, approaches that use only the textual content of the articles to predict the intended interpretation of ambiguous toponyms are more general and can be applied to a wider range of articles and other textual content.

Some works impose a grid on the earth and attempt to relate toponyms to geographic locations via cells in the grid [30]. The result is termed an *index* which is often implemented using hierarchical spatial data structures [23]. Rather than attempting to generate geographic coordinates, we instead focus on one toponym, Portland, and attempt to train an expert to resolve references to two different Portlands with known coordinates: the one in Maine and the one in Oregon. This simplifies the problem into a 2 class problem, which could easily be expanded to include additional Portlands besides the one in Maine and the one in Oregon, if need be. This would be useful for cities like Alexandria, where we could train a classifier to distinguish between Alexandria, Egypt, Alexandria, Virginia, and Alexandria, Louisiana.

We propose a method for training a Portland expert and evaluating its ability to accurately guess whether a mention of Portland in a news article refers to Oregon or Maine, given only the article text in which the mention is made. This is accomplished by training several models on weakly labeled data that was obtained using NewsStand's [15, 25–27, 29] rule-based geotagger [13, 14, 16, 17, 21] which is further discussed in Section 3. After achieving good results using NewsStand's interpretations of the article locations, we have begun the task of manually labeling the articles we gathered, in an effort to curate a ground truth dataset. We expect to use the ground truth labels to retrain our current models and improve performance.

We evaluate our models' performance on the weakly labeled NewsStand data and determine which models are robust to the removal of key indicator words from the articles. We show that a logistic regression model achieves good performance on the NewsStand dataset while remaining robust to the removal of a few key

LocalRec'21, November 2–5, 2021, Beijing, China

ACM ISBN 978-1-4503-9100-9/21/11…$15.00
https://doi.org/10.1145/3486183.3491066

indicator words from the dataset. We also perform feature reduction using NLP techniques. We demonstrate that leveraging Term Frequency Inverse Document Frequency (TF-IDF) to prune features improves recall on the class with fewer examples, without degrading the recall of the class with more examples.

While our approach may initially seem difficult to scale to address all duplicate toponyms, in reality, out of the over 12 million toponyms that exist, only about ten thousand are meaningful duplicates, based on our previous experience with NewsStand. This is a critical observation when considering our approach to the problem at hand. It means that to include other ambiguous toponyms we would not need an exponentially large number of classifiers or 'experts.' Instead, our observations of NewsStand data show that the true number of classifiers that would be needed is only a small fraction of the total number of toponyms that exist in the world. This is supported by [24] which sampled NewsStand articles at random and found that most of the time the toponyms were not ambiguous. This also matches their observation that Gazetteers have very few entries with multiple interpretations, and when they do, the number of interpretations is generally low. According to [14], *toponyms* and the *number of interpretations they have* exhibit a power-law relationship, meaning the vast majority of toponyms have a small number of interpretations, while a few toponyms have a very large number of interpretations. For these toponyms with many interpretations, like Paris or London, we suggest limiting the number of classes that would be needed in the classifier on the basis of population or other factors.

While it might also seem that there are many toponyms with similar names, in most cases there are meaningful differences that allow for easy disambiguation. This can be seen with 'York' and 'New York.' While these toponyms partially match, they can be easily differentiated by the existence of the word 'new.' This is a common schema in the naming of cities, where a word such as 'new' is appended onto the front of an existing city name in order to create another city's name. But in these cases the differences in the toponyms allow for clear interpretations of references to the relevant cities. Hence, to successfully disambiguate references to toponyms, we need only focus on exact duplicate toponyms, like Portland, Oregon and Portland, Maine. And since we know these are relatively few compared to the number of toponyms that exist, we find that it is not unreasonable to take the approach of developing an 'expert' model that can disambiguate toponym references for each meaningful duplicate name, like 'Portland.'

2 RELATED WORK

2.1 Using the Web to Resolve Toponyms and Vague Place References

It can be particularly difficult to associate an article's reference to a toponym to the actual intended location. There are a number of complicating factors that make this task challenging. In some cases crisp geographic boundaries are difficult to define, which necessitates the use of social or other attributes to define a location [2]. Work has also been done to classify land-use using bag-of-visual-words techniques with spatial extensions [31]. Even when the location is clearly definable, associating a reference to its intended location can be hampered by the noise implicit in the text of the article. And to add to the challenge, there are relatively few examples of labeled training data to work with. One approach taken by [21] is to exploit the link structure of wikipedia in order to generate local lexicons for different locations. They achieved an improvement on the disambiguation of toponyms with this method.

Likewise, [8] uses knowledge harvested from web pages in order to address the problems associated with vague references to place names. They find that vague place names are often accompanied by names of more precise co-located places that are located within the extent of the target vague place. These papers use outside information gleaned from the internet in order to geo-locate a toponym or a vague reference to a place.

For this study, we instead work with only the text of the article itself, and try to disambiguate toponym references based on the textual content in the article. We find that for at least one of the models, the important features that were used to decide whether an article referred to Portland, Maine or Portland, Oregon were often words that would qualify as being part of the local lexicon, which [21] defines as follows: "for a location s [local lexicon] is a set of concepts such as, but not limited to, names of people, landmarks, and historical events, that are spatially related to s." In particular, the features we find to be important to the decisions made by the trained models include geographically relevant words, such as locations that are nearby either Portland, Maine or Portland, Oregon. In this sense, the results of this study support the findings of [21], which state that the local lexicon is key to determining the spatial reader scope of news sources.

2.2 Using Gazetteers to Geotag Web Content

Amitay et. al [1] develop a system called Web-a-where to geotag web content. They attempt to disambiguate geo/geo ambiguities, which encompass situations where the same word refers to more than one distinct place. Our test case of Portland, Oregon vs. Portland, Maine falls into the category of geo/geo ambiguities. Their method, along with others like [19], rely on a gazetteer, which is a database containing geographic names and their canonical taxonomies. They scan the text of the web page for references to places that are known in the gazetteer, assign meaning to the reference, and then determine the focus of the entire web page based on the individual references made in the text.

2.3 Toponym Recognition

Toponym recognition is an area of research related to toponym resolution. Typically the tasks are done in sequence (recognition and then resolution), although some work has been done in separating and evaluating the tasks independently [14]. Leidner et. al [12] surveys methods for geoparsing (toponym recognition), which include the gazetteer method used in [1] as well as rule-based methods which attempt to match regular expressions and machine learning based methods that extract features from sliding windows run over the text. Rule-based methods are limited in their ability to expand to unusual formulations, i.e. wordings that don't match the regular expressions, and do not adapt well to changes in language over time and from place to place. Machine learning based geoparsing often involves using capitalization, length, and other features, which

are then checked for correlation with toponyms in order to decide whether or not a toponym has been found.

Our study deals with some of these principles from the perspective of toponym resolution, which happens once toponyms have been recognized. The use of capitalization in the English language tends to signify a word's importance, especially when referring to geographic locations- city, state, country names are usually capitalized. We show that our models are able to self-select features which represent locations as being more important than other words that appear in the articles. This is despite removing all capitalization during the preprocessing phase and providing the models no information about the meaning of the words themselves, apart from the frequencies with which they appear in each document.

2.4 Geoparsing Microtext

Geoparsers are also used for microtext such as social media posts [4–6, 18]. Social media posts differ from news articles in that they are much shorter, and often contain misspellings and abbreviations. This makes them a more challenging domain for toponym recognition and resolution. The method proposed in [6] finds place references that are abbreviated, misspelled, or highly-localized, since these are often missed by traditional geoparsers when applied to social media data. Their method identifies toponyms using gazetteer matching and Named Entity Recognition, and disambiguates abbreviations using a decision tree.

While we do not attempt to resolve toponyms in social media posts, it is interesting to consider how our model might fair on shorter text, as opposed to the longer news articles. Likewise, abbreviated words in an article are considered entirely separate features by our model, and hence also pose a potential challenge. The abbreviation disambiguation discussed in [6] could be applied to help with this issue, were we to adapt our method to shorter, more informal texts such as social media posts. The shorter length would also result in a sparser representation for each data point in our dataset, since we treat each individual article as a datapoint, and measure frequencies of each feature word in the article. Fewer words in the article (or post) would make for sparser representations, and this might affect our model's performance.

2.5 Toponym Resolution

Baldridge et. al [30] performs geolocation using a hierarchy of logistic regression classifiers. They attempt to map text to cells in a grid covering the earth. Their reason for choosing the hierarchy of logistic regression classifiers is because of the model's allowance of complex interdependent features. Although they find no noticeable improvement over bi-gram or character-based features, we find that logistic regression models performed the best at disambiguating Portland toponyms in NewsStand articles.

Further, in our experiments the logistic regression model learns alternate representations to make the classification between Portland, Oregon and Portland, Maine. Some of the other models show poor recall when all instances of the words "maine" and "oregon" are removed from the articles, whereas the logistic regression maintains its recall within 2%.

Baldridge et. al also provide the top 20 features selected for different regions on a 5° grid using their logistic regression. The types of words that they find in the top 20 for each reported location are similar in nature to the types of words that one of our models reports as being the most important features for decision making. That is, we both find that contextual words representing nearby locations (states and cities), as well as types of food are considered most useful for resolving toponyms.

For instance, we find that "seattle", "philly", and "canada" are important for resolving references to Portland, Oregon and Portland, Maine, and [30] finds that "idaho", "arizona" and "oakland" are important features when geolocating text that belongs to the Salt Lake City region, Phoenix region, and San Diego region, respectively. The fact that our findings follow the same trend indicates that it may be applicable across datasets. This is encouraging because it makes intuitive sense that references to nearby locations are good indicators for geolocating a text.

Other works also attempt to resolve toponyms to geographic coordinates [3, 10, 11]. Cardoso et. al [3] take a deep learning approach and use Recurrent Neural Networks (RNNs) to predict geographic coordinates of a toponym. They find that this approach outperforms the state of the art. Kulkarni et. al [10] also maps text to geographic coordinates, instead using a Multi-Level Geocoder. RNNs have also been used successfully in the related task of toponym matching, which involves detecting different strings that represent the same location in the real world [28].

We only attempt to disambiguate duplicate toponyms as referencing one of a fixed set of locations (i.e. Portland, Oregon or Portland, Maine) rather than narrowing to geographic coordinates as [3, 30] do or matching toponyms that refer to the same physical location as [28] does. Toponym resolution is also relevant in contexts outside of news articles. The work of [7] identifies and disambiguates toponyms in scientific text, using a toponymic search interface to correctly disambiguate 39% of place names referenced in 500 sentences from scientific texts.

2.6 Unsupervised Methods for Toponym Resolution

As [9] mentions, the issue of lack of labeled data is an "insurmountable" problem in toponym resolution. To address this, they take an unsupervised learning approach, modeling the cohesion of toponyms to context, as determined by geotagging. They apply a probabilistic model for disambiguation to each mention of a toponym in each article.

We avoid considering each mention of Portland within an article separately, and instead assume that all mentions within a single article are referring to the same Portland. In essence, we consider an entire article to have a single label, whereas [9] considers each instance of a toponym being mentioned to have its own label.

Although our approach simplifies the problem a bit which makes training easier, it adds noise because any cases where a single article refers to multiple toponym locations (i.e. refers to both Portland, Maine and to Portland, Oregon) will always be classified at least partially incorrectly by the weak labels for our dataset, and hence also by our model.

3 NEWSSTAND DATASET

The dataset of articles that we use for this study were geotagged by NewsStand [29]. NewsStand's interpretation of the references to Portland are at the article level, meaning that all references to Portland within one article are interpreted as referring to the *same* Portland, either Maine or Oregon. For instance, we assume that no article will use the word 'Portland' to refer Portland, Maine and then later in the article use the word 'Portland' to refer to Portland, Oregon. The way NewsStand makes its interpretations is described in detail in [14]. We use these interpretations as the 'labels' for a weakly supervised learning approach.

There are 257 articles that refer to Portland, Maine and 596 articles that refer to Portland, Oregon. Because the number of Portland, Oregon articles is more than double the number of Portland, Maine articles, we are also dealing with an unbalanced dataset, and need to take that into account when evaluating the performance of our Portland expert.

Being able to accurately determine when an article refers to Portland, Maine is just as important as being able to determine when one refers to Portland, Oregon, even though we have more examples of the latter. Hence, we evaluate the performance of each model in terms of its precision, and its recall on both the Oregon (OR) class and the Maine (ME) class. Not surprisingly, we find that in most cases the recall is better for the Oregon class, which has more samples than the Maine class.

3.1 Article Snippets

For some context, here are a few snippets of text that were extracted from the articles in this dataset.

3.1.1 Portland, Maine Snippets.

- "...In Portland, Mireille and Filipe describe their own arduous journey." ... "City Councilor Justin Costa told **Maine** Gov. Janet Mills at a public meeting last week that Portland still faces the long-term challenge" ... "the people of **Maine** have a 'proud tradition' of caring for their neighbors...."
- "...She's used to spending her days with students and colleagues at the University of Southern **Maine**, where she teaches sport management...."

3.1.2 Portland, Oregon Snippets.

- "...And she was a great player. I've heard stories that she gave **Seattle Storm** players all they could handle."
- "...researchers ... discovered commendable initiatives such as **Washington's** Clean Energy Omnibus Act..."

As is clear from the snippets above, there is ample context to indicate which Portland each article is referring to. In most cases that seems to come in the form of names of places or cities in either Maine or Oregon, or in neighboring states. This is the type of information that a Portland expert should learn to exploit, so that it can reliably disambiguate which Portland the articles are referring to.

In some cases, the words "maine" or "oregon" themselves show up in the article text. While this is an obvious clue that a Portland expert should exploit, we also test our models' ability to learn other representations of the two classes beyond just relying on the presence of the words "maine" or "oregon". To do this we remove any instance of the words "maine" or "oregon" in the articles and retrain the models. We find that the logistic regression and neural network models retain a high performance, demonstrating that they learned a more robust representation of the data.

4 METHODOLOGY

The general methodology follows the diagram in Figure 1.

4.1 Collecting the Data

NewsStand collects and geotags thousands of articles each day. Our machine learning based geotagging pipeline uses NewsStand as an article repository. In particular, articles are gleaned in one of following three ways. If an article contains a subset of the locations of interest then we add the articles to the training corpus. For example, if the article mentions Portland, Maine or Portland, Oregon, or any of the suburbs of these places we include the article in our dataset. We then trawl articles based on keywords, e.g., "Portland". Finally, we add articles of news websites whose source location is either of the Portlands of interest. Note that while we add the articles to the training corpus, we also retain the geotagged locations separately so that we can evaluate the performance of the algorithms later.

4.2 Data Preprocessing

To preprocess the data, we first read the dataset in and split it up into individual articles. We then remove any non-letters or special characters that appear in any of the articles. Next, we tokenize the articles into lists of words, and we strip punctuation from the tokens. Then, we convert all of the letters in the words to lower case, so that ultimately 'Washington!' is considered equivalent to 'washington'.

We also take two additional steps, which involve removing all stopwords using the Python NLTK library for English stopwords, and removing all instances of the word "maine" or the word "oregon" that appear in the article text. We perform tests with and without these final two steps and compare the performance in Section 5.

Once the data is cleaned, we split it into a training and a test set, and then calculate word frequencies for each article, and create a set of all words used in any of the training articles. This set of words, becomes the feature names to be used for training. And the frequencies for each word for each article become the feature values we train on. This is described in further detail in Section 4.4.

The list of words ultimately contained 37864 different words, including stopwords.

4.3 TF-IDF Analysis

To explore the dataset further, and figure out which words are considered important to the corpus of Portland, Maine texts and the corpus of Portland, Oregon texts, we performed Term Frequency Inverse Document Frequency (TF-IDF). The TF-IDF score we calculate for each word can be thought of as a measure of relevance of the word [22].

We rank the words based on their TF-IDF score, and show some examples of words that had high TF-IDF scores and words that had low TF-IDF scores below.

4.3.1 TF-IDF on the Portland, Maine corpus.

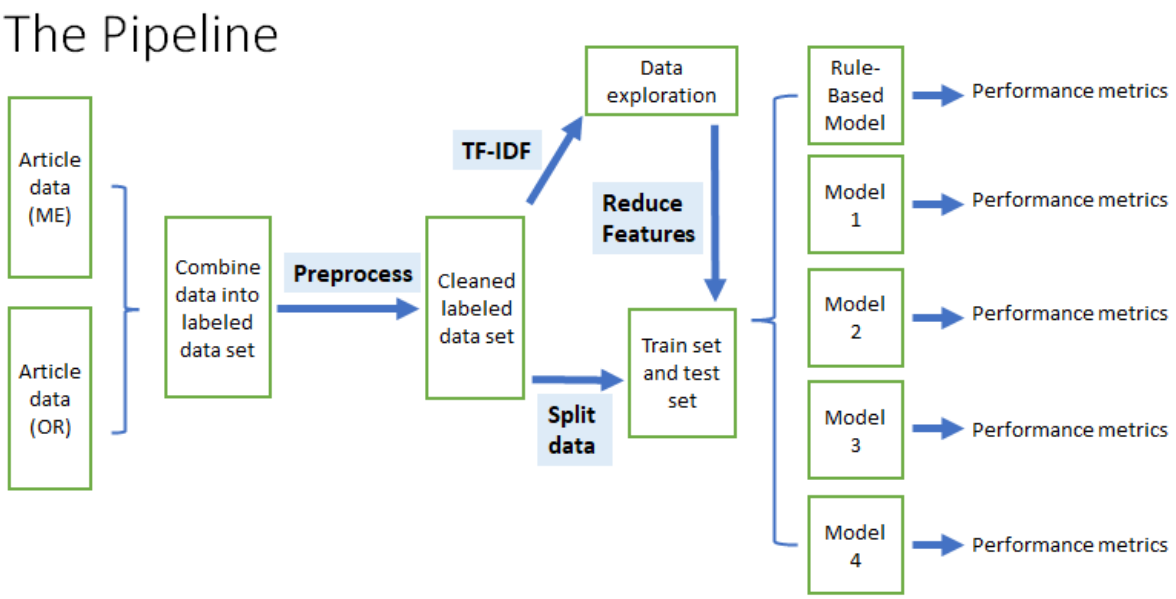

Figure 1: This is a diagram showing the high-level steps for training the 4 models on the NewsStand dataset of labeled articles.

- 'celtics': 0.12808559150077936
- 'vermont': 0.0816613930192776
- 'asbury': 0.05336185609159643
- 'westbrook': 0.050061122725105926
- 'lobster': 0.04958013147598997
- 'sugarloaf': 0.04316381248298022
- 'augusta': 0.04297245522511539
- 'appalachians': 0.04022011327319158
- 'massachusetts': 0.023062567678974598
- 'the': 0.0

4.3.2 *TF-IDF on the Portland, Oregon corpus.*

- 'wolves': 0.20286113531225947
- 'dallas': 0.15068231003389734
- salmon': 0.11794700674343546
- 'boulder': 0.06492795556315632
- 'denver': 0.05363216006804797
- 'twilight': 0.0529162837839724
- 'west': 0.04655204329646069
- 'washington': 0.007500214573585139
- 'seattle': 0.0059244911449860055
- 'portland': 0.0000166965844316579

Notably, words like 'the' and 'portland,' which we expect to show up in both types of articles and bear no impact on which Portland an article refers to, have a TF-IDF score of zero. This indicates that they are rather unimportant. On the other hand, words that do indicate the location of the article are rated as more important. This includes words that represent nearby locations, such as 'seattle,' a city that is near Portland, Oregon. Likewise, 'augusta', 'westbrook', and 'sugarloaf' are all located in Maine, and so it is no surprise that they have high TF-IDF scores for the Maine corpus.

We find that many of the words that ranked highly in terms of TF-IDF score here ended up also representing features that were important to the decisions made by the models we trained. We ultimately use the TF-IDF score to guide feature reduction, which is described in Section 5.2.

4.4 Word Frequencies

We treat the set of words that appear in the training articles as features in our dataset. Since we consider each article to be a datapoint, we measure how many times each of the feature words appears in a given article, and use those frequencies as the our feature values.

Below is a sample of some words along with their overall frequency counts for the entire Portland, Maine corpus, or the entire Portland, Oregon corpus.

4.4.1 *Sample of Word Frequencies from the Portland, Maine Corpus.*

- 'maine': 240
- 'massachusetts': 32,
- 'canada': 27,
- 'philadelphia': 14,
- 'shipbuilding': 4,
- 'acadia': 3,
- 'sherwood': 2,
- 'fayetteville': 2,
- 'philly': 2,

4.4.2 *Sample of Word Frequencies from the Portland, Oregon Corpus.*

- 'oregon': 255,
- 'mountain': 23,
- 'oregonian': 19,
- 'mount': 15,
- 'summit': 13,
- 'pioneers': 8,
- 'lumber': 7,

- 'washingtons': 6,
- 'seattles': 3,
- 'rainier': 3,

From the samples shown above, we can see that identifying words appear in the articles from both Portlands. For instance, using the word 'ranier' in an article is a good indication that it refers to Portland, Oregon, which is near the famous Mount Ranier. Likewise, mentioning 'acadia' usually refers to Acadia National Park which is a famous place in Maine, and so those articles probably refer to Portland, Maine and not Portland, Oregon.

After calculating the word frequencies for each article, we standardize the data and train several models.

4.5 Models

The set up we use for training models to discern between articles that refer to Portland, Maine and ones that refer to Portland, Oregon is as follows:

We take a supervised learning approach, using weakly labeled data that was labeled with NewsStand's geotagging system [29]. We consider words included in the text of the training articles to be features, and their number of occurrences in each article to be the feature values. Hence, each article is represented as a single data point. This means we have in total **257** data points for the Portland, Maine class and **596** data points for the Portland, Oregon class.

We train several models on this data set and evaluate the performance of each one on a hold out set. We take 90% of the articles in each class to be training data and reserve 10% from each class for testing purposes. This distribution is motivated by the fact that we have relatively few examples to work with and want to maximize the number of articles seen by the model during training. It should be noted that this approach to the problem is limited to the words that show up in the training articles, and the model will not be able to exploit words that are used only in the articles belonging to the test set that do not also appear in articles from the training set.

This problem is formulated as a 2-class classification problem. The class represented by 0 is Portland, Maine and the class represented by 1 is Portland, Oregon. The four types of models we train on out dataset are: Logistic Regression, Random Forest, Neural Network, and Support Vector Machine. For comparison, we list the performance of these models against a baseline "rule-based" model that guesses the article's location (Maine or Oregon) based on the existence of the word "maine" or the word "oregon" appearing in the article.

4.5.1 Logistic Regression. We first train a Logistic Regression model because it is simple, easy to implement, and fast to train. It can also be easily extended to a multi-class problem to predict more Portlands, if were to gather the appropriate data.

This model achieves a precision of 86.0% overall, and a recall of 73.9% on the Portland, Maine class and 90.5% on the Portland, Oregon class. This paradigm of a lower recall on the Portland, Maine class is to be expected, since we are dealing with an unbalanced dataset with significantly fewer examples of articles from Portland, Maine as compared to the number of articles from Portland, Oregon.

4.5.2 Random Forest. We also train a Random Forest, which is an ensemble classifier. This model is relatively simple and is fast during training and inference. It also provides easy visibility into which words were most important when classifying articles.

This model achieves an overall precision of 89.5% and a recall of 60.9% on the Portland, Maine class and a recall of 100.0% on the Portland, Oregon class. Again we see the same phenomenon of a lower recall on the Maine class.

Below we list a few of the most important words in deciding which place an article referred to:

- 'maine': 0.06356751296456024
- 'maines': 0.009135973781676097
- 'ore': 0.006979344701885978
- 'seattle': 0.0041547948426910895

Some other features that were less important than the ones above, but still appeared in the top 20 most important features were:

- 'oregon': 0.003400520305378662
- 'herald': 0.003394391785410089
- 'bangor': 0.0025408513933448383

Clearly, some of these words seem like good indicators of which Portland an article refers to. For example, if an article mentions Portland and 'Seattle', the Portland it refers to is probably Portland, Oregon, which is closer to Seattle. Likewise, 'Bangor' is much closer to Portland, Maine- in fact it is located in the state of Maine. Hence, an article that refers to Bangor is intuitively more likely to refer Portland, Maine than to Portland, Oregon.

Based on these important features, it seems like the Random Forest is classifying articles based on words that are intuitively good indicators of which Portland an article is referring to. And many of these words overlap with the words that TF-IDF rated as important words, which is a good indication that the model is exploiting the important words and generally ignoring the noise from the unimportant words.

4.5.3 Neural Network. The third model we train is a Multilayer Perceptron, also known as a Neural Network. This is a more complex model than the two previous ones, which makes it better suited to learn more nuances in the dataset. However, this comes at the price of a longer training time, and the need for significant hyperparameter tuning in order to achieve a reasonable performance. Neural networks are also prone to overfitting, which is an issue for our problem in particular because we have more features than data points. With minimal hyperparameter tuning the following architecture is adopted:

- Number of Layers: 3
- Number of nodes per layer: 50, 20, 10
- Learning rate scheme: adaptive
- Initial learning rate: 0.05
- Number of Epochs: 500

This model achieves an overall precision of 84.9% and a recall of 73.9% on the Portland, Maine class, and a recall of 88.9% on the Portland, Oregon class. Even with significant feature reduction (described in section 5.2) the Neural Network does not improve significantly beyond these results.

4.5.4 Support Vector Machine. The final model that we train on this dataset is a Support Vector Machine (SVM). This model finds a

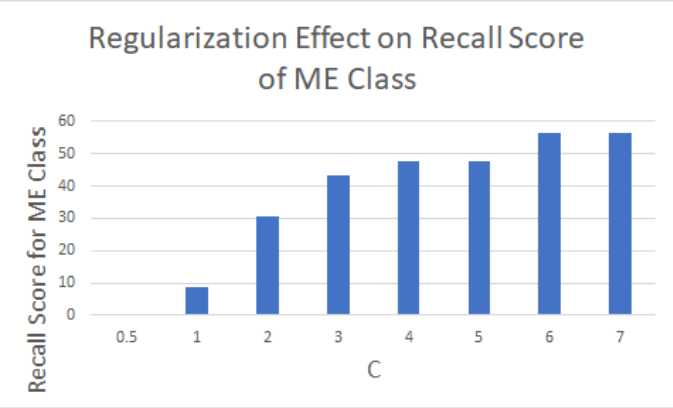

Figure 2: This chart shows how the recall score of the Portland, Maine class improves as we tune the *C* parameter of the SVM model.

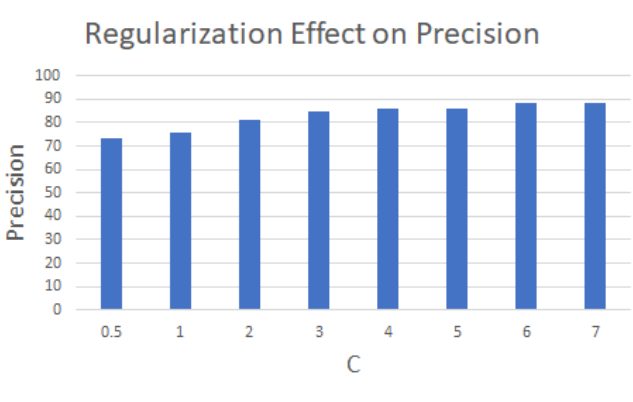

Figure 3: This chart shows how the overall precision score improves as we tune the *C* parameter of the SVM model.

separating hyperplane to distinguish between the 2 classes- Portland, Maine and Portland, Oregon. We chose this model because it can deal with very high dimensional spaces, which have caused significant overfitting in some of the models described above.

The SVM model initially achieves an overall precision of 73.2% and a recall of 0.0% on the Portland, Maine class and a recall of 100.0% on the Portland. Oregon class. Clearly, this performance is due to the model just guessing that every article is referring to Portland, Oregon (based on the recall of 0% for the Maine class and 100% for the Oregon class). In order to try to improve this performance, we tune the amount regularization the model uses. We adjust the squared L2 penalty term by changing the parameter *C*, which is inversely proportional to the strength of the L2 penalty.

Table 1: Tuning L2 Regularization Penalty Coefficient

Value of C	Precision	Recall (Maine)	Recall (Oregon)
0.5	73.3%	0.0%	100.0%
1.0	75.6%	8.7%	100.0%
2.0	81.4%	30.4%	100.0%
3.0	84.9%	43.5%	100.0%
4.0	86.0%	47.8%	100.0%
5.0	86.0%	47.8%	100.0%
6.0	88.4%	56.5%	100.0%
7.0	88.4%	56.5%	100.0%

Note, we can see in Table 1 and the corresponding chart in Figure 2 that tuning the *C* value results in a significant improvement in the recall score for the Maine class. Although this model still shows imbalanced performance, we improved the recall from 0.0% to 56.5% by changing only the extent of regularization used in the model. Precision also improved modestly, as we can see in Figure 3.

After tuning, this model achieves an overall precision of 88.4% and a recall of 56.5% on the Portland, Maine class and a recall of 100.0% on the Portland, Oregon class. This is a large improvement over the untuned performance listed above (overall precision of 73.2% and a recall of 0.0% on the Portland, Maine class and a recall of 100.0% on the Portland. Oregon class). However, the recall of this model remains poor compared to the other models we trained.

4.5.5 Rule-Based Model. As a baseline model, we use a simple rule that if an article contains the word "maine" then it is labeled as the Maine class, and if it contains the word "oregon" it is labeled as the Oregon class. When neither word or both words appear in a given article, this model fails. We report the overall precision, as well as the recall on each class for this model as a baseline for comparison with the four models described above.

5 RESULTS

5.1 Performance by Model

5.1.1 Overall Performance. Table 2 reiterates the performances listed above for each of the models, in a more condensed format. For the SVM model, we consider the best performance after tuning the regularization parameter.

Table 2: Performance on Test Set (with stopwords and 'Maine', 'Oregon' left in)

Model	Precision	Recall (ME)	Recall (OR)
Baseline (Rule-Based)	64.2%	89.1%	39.9%
Logistic Regression	86.0%	73.9%	90.5%
Random Forest	89.5%	60.9%	100.0%
Neural Network	84.9%	73.9%	88.9%
SVM	88.4%	56.5%	100.0%

All four models we trained show significant improvement in precision over the rule-based model used as a baseline, with the Random Forest having the highest precision overall. In terms of recall for each class, we see that the 4 models outperform the baseline substantially on the Oregon class, which had a large number of training examples, whereas the rule-based model does the best on the smaller Maine class. We can see that the Logistic Regression does the best job at achieving a balance between the two classes, despite the Maine class having less than half of the available training

examples compared to the Oregon class. **Hence, our best model overall is the Logistic Regression.**

5.1.2 Removing Key Indicator Words. We then test how the model performance changes when the key indicator words "maine" and "oregon" are removed from the articles during preprocessing. Models that relied on those words to classify the articles without learning other representations of each class will perform poorly now, whereas models that learned a more robust representation of the Maine class and the Oregon class will see little change in performance. This is particularly interesting given the fact that the Random Forest model ranked the words "maine" and "oregon" as being very important features for decision making (i.e. they appeared frequently enough in the articles to be useful when distinguishing between the two classes). The results are described in Table 3.

Table 3: Performance on Test Set (with stopwords left in and 'Maine', 'Oregon' removed)

Model	Precision	Recall (ME)	Recall (OR)
Baseline (Rule-Based)	0.0%	0.0%	0.0%
Logistic Regression	86.0%	73.9%	90.5%
Random Forest	83.7%	39.1%	100.0%
Neural Network	87.2%	78.3%	90.5%
SVM	88.4%	56.5%	100.0%

With the removal of the words "maine" and "oregon" from the articles, our baseline rule-based model fails to make any predictions. However, we find that the performance of the logistic regression remains the same and the performance of the neural network increase slightly. This indicates that these two models are able to classify the news articles based on contextual information aside from the key indicator words "maine" and "oregon". The other models (Random Forest and SVM) see a large drop in performance upon removal of these two features, indicating that they relied heavily on those features to make classifications. **Hence, two of our models, the logistic regression and the neural network, have high performance and are also robust to the removal of key indicator words "maine" and "oregon" from the text.**

5.1.3 Removing Stopwords. We then test how reducing the number of features by removing stopwords affects the performance of the models. Since stopwords are very common and should appear equally frequently in articles referring to Portland, Maine and articles referring to Portland, Oregon, removing them should not strongly impede performance. The results are described in Table 4.

As with the previous experiment, the rule-based model used as a baseline fails to make any predictions once the key indicator words "maine" and "oregon" are removed from the text, regardless of the existence or removal of stopwords. For the logistic regression, after removing stopwords we find a drop in precision of about 1.1% and a drop in recall for the Oregon class of about 1.6%. The other models show substantially reduced performance on the Maine class (the class with fewer training examples) after stopwords are removed. This may be due to overfitting. **These experiments show that the logistic regression model is the most robust overall, both to the removal of key indicator words, and to the existence or removal of stopwords**.

Table 4: Performance on Test Set (with stopwords and 'Maine', 'Oregon' removed)

Model	Precision	Recall (ME)	Recall (OR)
Baseline (Rule-Based)	0.0%	0.0%	0.0%
Logistic Regression	84.9%	73.9%	88.9%
Random Forest	80.2%	26.1%	100.0%
Neural Network	75.6%	13.0%	98.4%
SVM	88.4%	56.5%	100.0%

5.2 Feature Reduction

Table 5: Performance on Test Set After Feature Reduction

Models are abbreviated as follows: Logistic Regression (LR), Random Forest (RF), Neural Network (NN), Support Vector Machine (SVM)

Min TF-IDF score (θ)	Num feats kept	Model	Precision	Recall (ME)	Recall (OR)
0.0	37862	LR	86.0%	73.9%	90.5%
		RF	83.7%	39.1%	100.0%
		NN	87.2%	78.3%	90.5%
		SVM	88.4%	56.5%	100.0%
0.001	35841	LR	86.0%	73.9%	90.5%
		RF	81.4%	30.4%	100.0%
		NN	73.3%	0.0%	100.0%
		SVM	88.4%	56.5%	100.0%
0.01	24864	LR	84.9%	73.9%	88.9%
		RF	83.7%	39.1%	100.0%
		NN	67.4%	73.9%	65.1%
		SVM	87.2%	56.5%	98.4%
0.05	**19372**	**LR**	**87.2%**	**78.3%**	**90.5%**
		RF	81.4%	34.8%	98.4%
		NN	87.2%	78.3%	90.5%
		SVM	87.2%	52.2%	100.0%
0.1	19019	LR	83.7%	69.6%	88.9%
		RF	81.4%	30.4%	100.0%
		NN	73.3%	0.0%	100.0%
		SVM	88.4%	60.9%	98.4%

Because our approach involves treating each word that appears in the training set of articles as its own feature, we inherently have a large number of features. This is not ideal given the small number of samples in our dataset (less than 1000 articles total). Hence, we perform feature reduction to deal with the curse of dimensionality.

We use the TF-IDF scores calculated for each word in the training set in order to decide whether or not a given word is important enough to be used as a feature. Words that have a score above a minimum threshold θ are kept as features and the remaining

features are thrown out. We show the performance of each model on the reduced set of features for various θ in Table 5.

We find that reducing the number of features using a threshold of $\theta = 0.05$ produces the best performance in terms of precision and recall on both classes for the logistic regression and the neural network models. This threshold yields a feature reduction from 37862 features to 19372 features. For our most robust model, the logistic regression, the resulting performance changes are a 1% increase in precision and 4% increase in recall on the Maine class, which was the class that had lower recall originally. Notably, there was no reduction in recall for the Oregon class. This indicates that the use of TF-IDF to inform feature reduction was successful at eliminating noisy features while retaining the important ones, and boosting performance overall.

6 LIMITATIONS

There are three major limitations of this work. The first has to do with the size of the dataset used. Since we only had between 250 and 600 articles per class, and we split the dataset to include 10% of each class in the test set, we only tested the models on between 25 and 60 articles per Portland. This is too few articles to really know how well the models perform, and so the models should be evaluated on a larger set of data to verify the validity of the approach taken.

The second limitation is that the models here do not account for the meaning of the words in the articles. Although analysis shows that the words that at least one of the models (the Random Forest) selects are intuitively good indicators of location, the models do not intrinsically have any linguistic information to help make decisions. To remedy this, we might use more NLP techniques to introduce information about topic or parts of speech, which may also help improve performance.

And finally, the most significant limitation of this approach is that the models are trained on a static set of articles that was collected over a relatively short period of time. Since the models assume that articles seen at training time are representative of all articles mentioning Portland, there is the obvious issue that language changes over time. As events occur and things change in each city, so too will the language of the articles being written about each place. Our models and any that are trained on this dataset will fail to capture those changes, and hence may generalize poorly to future articles about the two Portlands.

To underscore the issue here, we can consider an example. The word 'coronavirus' was one of the relatively important (although not most important) words that the Random Forest model used to discriminate between Portland, Maine and Portland, Oregon. If we consider that COVID-19 was detected in Oregon before Maine, it is plausible that for a time more articles about coronavirus would refer to Portland, Oregon than Portland, Maine. However, COVID-19 eventually did spread across the United States, and so it is also reasonable that later articles may have discussed Portland, Maine in relation to coronavirus. But if the articles in this dataset were all collected between the time that the virus struck Oregon and the time that it spread to Maine, the models we train might learn that the word 'coronavirus' is a strong indicator that 'Portland' refers to Oregon, but for a more recent population of articles mentioning Portland, this assumption may be invalid. Hence, if our models learned this association between 'coronavirus' and Portland. Oregon, they may not generalize well.

This phenomenon of changing language is a major challenge in this area of toponym resolution, because it necessitates constant retraining of models, which is difficult given the scarcity of labeled datasets in the first place. A different approach to solving or at least offsetting this problem would be to encourage the models to learn associations that are more static, like words that refer to nearby cities, since those names will probably not change. In other words, it is safer for a model to rely on 'Seattle' being an indicator of Portland, Oregon than it is to rely on 'coronavirus' being an indicator of Portland, Oregon, since the first association is unlikely to change over time, whereas the second is *event driven* and hence more likely to vary over time.

7 CONCLUSIONS

In general, all four of the models we trained outperformed the baseline rule-based model in terms of overall precision, with the logistic regression performing the best. Averaging recall over the two classes, we achieved at best mid 80% range. For the class with more training examples, Oregon, our models achieved substantial improvement over the rule-based model used as a baseline. This indicates that the models learned contextual information aside from the words "maine" and "oregon" to resolve the toponym references. For the class with fewer training examples, Maine, the models performed worse in terms of recall. This can most likely be attributed to the fact that the Maine class had less than half the number of examples compared to the Oregon class. This means that there was less contextual data about Portland, Maine for the classifiers to learn from, which likely hindered the performance on this class.

When testing the models' ability to resolve toponyms with any instance of the words "maine" or "oregon" removed, the logistic regression maintained its performance, indicating that it learned contextual information about the two locations (Portland, Maine and Portland, Oregon). This is a particularly important characteristic for a classifier to have in this problem domain because we are specifically interested in resolving cases of toponyms that lack obvious clues about which location they refer to. These are the cases where rule-based methods, like our baseline, fail to accurately disambiguate the toponym reference, and so a machine learning classifier that learns from less obvious contextual clues adds the most value in these cases.

The words that the models relied on seem to intuitively make sense as differentiators between Portland, Maine and Portland, Oregon. These words also aligned well with the words that TF-IDF suggested were important to both sets of data. When performing feature reduction, we exploited the TF-IDF scores assigned to each word in order to prune features. We tested several TF-IDF score thresholds and found that a score of 0.05 was the optimal threshold, providing an improvement in the recall of the smaller class (Maine) of about 4%, and an improvement in overall precision of about 1%, without causing a reduction in the recall score of the Oregon class.

8 FUTURE WORK

There are several avenues of future work that could improve upon this study. The first is to incorporate some other NLP techniques

to help augment the models' ability to learn the problem. These might include topic modelling, part of speech tagging, stemming, and more. This could also include doing more processing and NLP upfront, to help eliminate some of the noise in the dataset. Feature engineering could also be done to combine features and reduce the overall dimenstionality of the problem.

Secondly, we might train and test the models on more articles. While the results we obtained show promise for the techniques presented in this paper, a wider range of articles would be useful for improving the generalizability of our models over time. Periodic retraining of the models with new data may be necessary in order to overcome the fact that language changes over time, and any model is only as good as the data it sees during training. This highlights the fundamental issue with using a static corpus of data, which is an issue that NewsStand is well positioned to address.

Gathering more articles for Portland, Maine may be particularly helpful in improving performance on that class, since we only had about 257 articles to train on for that class. From our results, it seems the 597 articles for Portland, Oregon was sufficient to learn that class well. It would also be helpful if the additional articles came from a wide range of time periods, that way we could begin to address the generalization across time issue discussed in section 6. Without additional articles, techniques like oversampling or undersampling could be used to address the unbalanced dataset.

We are also currently undertaking the manual labeling of the NewsStand dataset that was used to train our models. This will allow us to test the performance of the models without relying as heavily on the performance of NewsStand's geotagger which provided the initial labels that were used during training.

ACKNOWLEDGMENTS

This work was sponsored in part by the NSF under Grants IIS-18-16889, IIS-20-41415, and IIS-21-14451. We also thank Jagan Sankaranarayanan for the study idea and the data sets provided to train the models.

REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. 2004. Web-a-Where: Geotagging web content. In *Proceedings of SIGIR'04*. Sheffield, United Kingdom, 273–280.

[2] T. Blaschke, H. Merschdorf, P. Cabrera-Barona, S. Gao, E. Papadakis, and A. Kovacs-Györi. 2018. Place versus space: from points, lines and polygons in GIS to place-based representations reflecting language and culture. *ISPRS International Journal of Geo-Information* 7, 11 (2018).

[3] A. B. Cardoso, B. Martins, and J. Estima. 2019. Using recurrent neural networks for toponym resolution in text. In *Proceedings of the EPIA Conference on Artificial Intelligence*. Springer, 769–780.

[4] Z. Cheng, J. Caverlee, and K. Lee. 2013. A content-driven framework for geolocating microblog users. *ACM Trans. Intell. Syst. Technol.* 4 (2013), 2:1–2:27.

[5] T. Fornaciari and D. Hovy. 2019. Identifying linguistic areas for geolocation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Association for Computational Linguistics, Hong Kong, China, 231–236.

[6] J. Gelernter and S. Balaji. 2013. An algorithm for local geoparsing of microtext. *GeoInformatica* 17, 4 (Oct. 2013), 635–667.

[7] X. Jiang and V. I. Torvik. 2020. On the ambiguity and relevance of place names in scientific text. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (Virtual Event, China) *(JCDL '20)*. Association for Computing Machinery, New York, NY, USA, 401–404.

[8] C. B. Jones, R. S. Purves, P. D. Clough, and H. Joho. 2008. Modelling vague places with knowledge from the Web. *International Journal of Geographical Information Science* 22, 10 (2008), 1045–1065.

[9] E. Kamalloo and D. Rafiei. 2018. A coherent unsupervised model for toponym resolution. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1287–1296.

[10] S. Kulkarni, S. Jain, M. J. Hosseini, J. Baldridge, E. Ie, and L. Zhang. 2020. Spatial language representation with multi-level geocoding. *arXiv preprint arXiv:2008.09236* (2020).

[11] J. L. Leidner. 2004. Toponym resolution in text: "Which Sheffield is it?". *Proceedings of SIGIR'04* (July 2004), 602.

[12] J. L. Leidner and M. D. Lieberman. 2011. Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Special* 3, 2 (2011), 5–11.

[13] M.D. Lieberman and H. Samet. 2011. Multifaceted toponym recognition for streaming news. In *Proceedings of SIGIR'11*. Beijing, China, 843–852.

[14] M. D. Lieberman and H. Samet. 2012. Adaptive context features for toponym resolution in streaming news. In *Proceedings of SIGIR'12*. Portland, OR, 731–740.

[15] M. D. Lieberman and H. Samet. 2012. Supporting rapid processing and interactive map-based exploration of streaming news. In *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Redondo Beach, CA, 179–188.

[16] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. 2010. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *Proceedings of 6th Workshop on Geographic Information Retrieval*. Zurich, Switzerland.

[17] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proceedings of the 26th IEEE International Conference on Data Engineering*. Long Beach, CA, 201–212.

[18] F. Liu, M. Vasardani, and T. Baldwin. 2014. Automatic identification of locative expressions from social media text: A comparative analysis. In *Proceedings of LocWeb'14*. Shanghai, China, 9–16.

[19] B. Martins, H. Manguinhas, and J. Borbinha. 2008. Extracting and exploring the geo-temporal semantics of textual resources. In *Proceedings of ICSC'08*. Santa Clara, CA, 1–9.

[20] G. Quercini and H. Samet. 2014. Uncovering the spatial relatedness in Wikipedia. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Dallas, TX, 153–162.

[21] G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman. 2010. Determining the spatial reader scopes of news sources using local lexicons. *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 43–52.

[22] J. Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, Vol. 242. 29–48.

[23] H. Samet. 1990. Hierarchical spatial data structures. In *Design and Implementation of Large Spatial Databases*. Springer Berlin Heidelberg, Berlin, Heidelberg, 191–212.

[24] H. Samet. 2014. Using minimaps to enable toponym resolution with an effective 100% rate of recall. In *Proceedings of GIR'14*. Dallas, TX.

[25] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and B. E. Teitler. 2011. Porting a web-based mapping application to a smartphone app. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Chicago, IL, 525–528.

[26] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowski, D. Panozzo, J. Sperling, and B. E. Teitler. 2014. Reading news with maps by exploiting spatial synonyms. *Commun. ACM* 57, 10 (Oct. 2014), 64–77.

[27] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. 2011. Adapting a map query interface for a gesturing touch screen interface. In *Proceedings of the Twentieth International Word Wide Web Conference (Companion Volume)*. Hyderabad, India, 257–260.

[28] R. Santos, P. Murrieta-Flores, P. Calado, and B. Martins. 2018. Toponym matching through deep neural networks. *International Journal of Geographical Information Science* 32, 2 (2018), 324–348.

[29] B. Teitler, M. D. Lieberman, J. Sankaranarayanan D. Panozzo, H. Samet, and J. Sperling. 2008. NewsStand: A new view on news. *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 144–153.

[30] B. Wing and J. Baldridge. 2014. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Doha, Qatar, 336–348.

[31] Y. Yang and S. Newsam. 2010. Bag-of-visual-words and spatial extensions for land-use classification. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 270–279.