Identifying Destinations Automatically from Human Generated Route Directions

Xiao Zhang† Prasenjit Mitra*, Alexander Klippel‡ Alan M. MacEachren*  
*Department of Computer Science and Engineering  
†College of Information Science and Technology  
‡Department of Geography  
the Pennsylvania State University  
xiazhang@cse.psu.edu, pmitra@ist.psu.edu, klippel@psu.edu, maceachren@psu.edu

ABSTRACT
Automatic and accurate extraction of destinations in human-generated route descriptions facilitates visualizing text route descriptions on digital maps. Such information further supports research aiming at understanding human cognition of geospatial information. However, as reported in previous work, the recognition of destinations is not satisfactory. In this paper, we show our approach and achievements in improving the accuracy of destination name recognition. We identified and evaluated multiple features for classifying a named entity to be either ‘destination’ or ‘non-destination’; after that, we use a simple yet effective post-processing algorithm to improve classification accuracy. Comprehensive experiments confirm the effectiveness of our approach.

Categories and Subject Descriptors
H.3.3 [INFORMATION STORAGE AND RETRIEVAL]: Information Search and Retrieval

Keywords
destination name classification, geospatial information extraction, driving directions

1. INTRODUCTION
Human-generated route directions are text descriptions of routes, which give detailed step-by-step instructions to travelers to reach a specific physical location from a starting point or region. They contain rich expressions of movement patterns, landmarks and decision points, etc. They have drawn the attention from researchers in the research areas of spatial information processing, human perception of spatial information and linguistics [5, 3].

In order to automatically geocode and interpret human-generated route directions, accurate information extraction must be performed. Among all types of useful information, we focus on extracting the endpoints of routes, that is, destinations. Destinations are potentially the most critical part in route directions. Endpoints are regarded as critical in the conceptualization of events [10] but they are also essential in reasoning about other linguistically encoded information in route directions [6].

Previous work on identifying route information from direction documents [12] extracts route components at a sentence level, i.e., the whole sentence containing the destination is extracted. Without knowing what exactly the destinations are, this information will be hard to use for geocoding. Besides, the recognition accuracy for destination sentences is not satisfactory, as reported. In [4], the authors used rule-based method to extract place names and geocode them. The route is then recovered by connecting the geocoded places on the map. However, it does not distinguish destinations from other landmarks and decision points along the routes. The best guess, one can make to find the destination, is to use the last place name as the destination. However, the locations where destinations appear in the text is far less regular than the other route information [13]. The lack of automatic and accurate recognition systems leave the heavy burden of identifying the destinations to human annotators.

Destinations are frequently referred to by their names in the text, for example, “College of Information Science and Technology” and “Saint Paul’s Church”. Destination names are a particular type of named entities. They co-exist with other named entities such as road names, place names and landmarks in the route direction documents. Due to the high accuracy of existing named entity recognition methods and systems (see [11] and [8] for surveys on NER), we focus our study on the unique characteristics of destination names and solve the destination name recognition problem by applying named entity classification techniques. First, we exploit existing named entity recognition systems to extract all named entities from route direction documents. Then, we build a binary classifier to classify the extracted named entities to be either “destinations” or “non-destinations”. After that, we apply a post-processing algorithm, which re-labels un-recognized destinations according to their similarity to recognized names, to improve recognition accuracy. When solving the binary classification task, we explore various feature sets that use general knowledge about what could be a destination, syntactic features, multiple data sources such as domain URL registrants’ names and online phone books, and so forth. Using our approach increases the accuracy with which destinations can be identified automatically.

2. PRELIMINARIES AND BACKGROUND
Human-generated driving direction documents contain the following route components [12]: destinations, origins and route parts; in addition, such a document may also contain information irrelevant to driving directions, such as advertisements. Figure 1 shows an example, with the three route components extracted. A typical
direction document contains only one destination and 3 - 5 origins. For each origin, a set of instructions is provided. We focus on this type of directions.

**Figure 1:** An example of a human-generated driving direction document

Among the various information present in the driving direction document, we are particularly interested in one type of named entities - destination names (defined in Section 3). Destination names can be presented in quite different ways: some are in enlarged fonts, highlighted with bright colors, with strong semantic clues such as "Driving directions to ...". Some are buried in a large paragraph of text with many other landmarks and place names where no obvious visual features make them stand out from the other named entities. Some do not even have their full names explicitly mentioned and are only referred to by their types, such as "the school". The destination names can appear multiple times and in multiple parts of the document, such as the title, the beginning of the document body, the beginning of a paragraph, a route instruction sentence, navigational link, or the footnote of the page. Various ways of presenting the destination names increase the difficulty of automatic recognition of a destination in a document.

Destination names often can be found in two types of sentences: (1) destination sentences [12] and (2) arrival information. Destination sentences explicitly specify the destination by mentioning its name, for example, “Direction to ISE Department of Rutgers” and “Visiting CASE”. Arrival information sentences are often one of the last few sentences in a set of route instructions. The destination names or the type of place, such as "the Center", "the Church", are often mentioned. For example: “The Center is located 0.4 miles on the left”. Obvious language patterns can be found to help us identify the destination names, such as “Directions to ..." and "... is located ...", etc.

The destination is often the main subject of the web site where the document resides. This leads to the following three important observations: (1) the provider of the driving directions is often the owner of the web site and thus the registrant of the web site’s domain URL. (2) The destination name, may also appear in the body and/or title of other web pages of the same site. (3) A phone number, if present, often belongs to the destination company or organization.

In Section 4, we will see that the above observations and analysis result in important features for finding destination names.

### 3. PROBLEM STATEMENT AND ANALYSIS

We first define the following important terms:

**Definition 1 (Destination).** A destination in a route direction document is a physical location where the route ends.

**Definition 2 (Destination Name).** A destination name in a route direction document is a named entity referring to the destination of the routes, usually an organization, company or office. It can also be the building or place name where the destination resides.

More than one name can be used to refer to the same destination in the document. For example, "Emory University Campus" is referred to as "Emory University Campus", "Emory Campus" and "Emory" in the same document. These are called "variations" of the destination name. Abbreviations or nouns or noun phrases of the type of the destination can also be used. For example, "Peters Township High School" is referred to as "PTHS" and "the school". The name of the destination, its variations, abbreviations and type nouns, are all considered to be destination names.

**Problem Statement 1.** Given a human-generated driving direction document, our task is to identify all destination entities: the proper names, variations, abbreviations and the type nouns referring to the destination.

Destination entities are organization or location named entities. However, only recognizing all organization and location named entities does not suffice to solve our problem because named entities other than destinations frequently appear as landmarks. For example, in "...go about one-third mile to the first traffic light, by the Home Depot.", "Home Depot" is a landmark to help travelers locate themselves and make decisions. Therefore separating the destination names from other named entities is an important task.

We solve the destination name recognition problem in two steps. First, named entities are extracted from a document. Second, each extracted named entity is classified as either "destination" or "non-destination". The choice of the distinguishing features is a critical design step and requires comprehensive study of the problem domain. Therefore, we put our effort on finding good features for recognizing destinations. As will be shown later, we identified and evaluated a set of useful features for our classification task. In addition, we analyzed the obtained experiment results. Some destination names were not recognized by the classification model. However, using the recognized destination names from the classification results, we are able to recognize some mis-classified destination names. For example, if "Emory University campus" is classified as "destination", then "Emory campus" is highly likely to be a "destination" because the two names are similar. Cosine Similarity captures this kind of similarity. A rule-based post-processing algorithm was developed to improve the classification accuracy. Section 4.2 gives the details.

### 4. METHODS

We solve the problem defined above in three steps: first, extract named entities from the document, using the algorithm described in previous work [12] and OpenCalais [1]; second, classify each extracted named entity to be either "destination" or "non-destination"; and finally, based on the classification results, re-classify a named entity according to its cosine similarity to a classified destination, as a post-processing step.

#### 4.1 Destination Name Classification

We built a binary classifier and explored extensively the relevant features to classify these named entities to be either "destination" or "non-destination". Depending on the sources of the information, we divide the 8 feature sets into two categories: feature sets 1 - 5 are extracted by analyzing the named entity itself and the document
containing it; feature sets 6 - 8 are obtained by linking the information in the document with external information sources, such as online phone book search engines and domain registrant databases.

**Feature Set 1** captures the “shape” of the named entity, including: whether the name has (1) all letters of all terms capitalized, or (2) all initial letters of all terms capitalized, or (3) length less than or equal to 5.

**Feature Set 2** matches the sentences containing the named entity against 9 sets of pre-defined language patterns, such as “... is located on your right” and “... will be the second building on the left”.

**Feature Set 3** checks if the named entity contains a type noun or a US state name. We identified a list of 129 destination type nouns, such as “school”, “university” and “hotel”. These are the type of businesses and organizations providing the driving directions.

**Feature Set 4** includes HTML visual features: route direction authors frequently use HTML visual features to emphasize route components. For example, destinations often appear in the titles and headings of the web page. Our approach checks whether a named entity is in the title, heading, or is a link.

**Feature Set 5** is the normalized count. Destinations often appear more than once in the document. We calculated the count and normalize by document length. If the normalized count is larger than a threshold \( n_c \), we assign the feature to this candidate. We evaluated different values for the threshold, from 0 to 1 with step size 0.1. We used the same set of values on the following thresholds.

**Feature Set 6**: Domain registrant’s names. Given the URLs of the direction web pages, we look up the URLs to obtain the domain registrants’ names, then calculated the cosine similarity between the extracted names and the registrant names. If the similarity exceeds a predefined threshold \( d \), we assign this feature to the named entity.

**Feature Set 7**: Phone book search results. If phone numbers are present, we look them up in the online phone book search engine to get the names associated with each phone number. Then calculate the cosine similarity similar between the obtained name and the named entity. If it exceeds a pre-defined threshold \( p \), we assign the feature to the named entity.

**Feature Set 8**: We crawled the titles of other web pages in the same domain as the driving direction page, and computed the proportion of titles containing the extracted named entity. If the proportion exceeds a pre-defined threshold \( t \), we assign the feature to the named entity.

### 4.2 Post-Processing for Improving Destination Name Recognition

After training and evaluation, we found that some destination names were mis-classified as “non-destination” due to lack of strong features: some names only appeared once in the document; they do not have strong language pattern features, such as “directions to ...”; they do not match any registered names from external information sources. However, they are similar to some of the correctly classified destination names. For example, “Christiana Hospital” was classified as “non-destination”; while “Christiana Hospital campus” was classified as “destination” correctly. Based on the observation, we designed a post-processing step to improve the recognition of destination names. For each document, we take the named entities classified as “non-destination” and calculate the cosine similarities between this named entity and all the named entities classified as “destination”. If one of the similarities is larger than or equal to a pre-defined threshold \( c \), we re-classify the named entity as “destination”. Algorithm 1 gives the details.

---

**Algorithm 1 Post-Processing**

**Input:**

(1) Pairs of named entities and predicted class labels from one document: \((e_1,l_1),(e_2,l_2),...,\(e_n,l_n)\) where \(l_i \in \{\text{destination}', \text{non-destination}'\}\)

(2) A threshold for cosine similarity comparison \( c \)

**Output:**

Pairs of named entities and their new class labels: \((e_1,l_1'),(e_2,l_2'),...,\(e_n,l_n)\)

**Procedure:**

1. for each \((e_i,l_i)\), where \(l_i \neq \text{destination}'\) do
2.   for each \((e_j,l_j)\) where \(j \neq i\) and \(l_j = \text{destination}'\) do
3.     \(sim \leftarrow \text{cosine_similarity}(e_i,e_j)\);
4.     if \(sim \geq c\) then
5.       \(l_i \leftarrow \text{destination}'\);
6.     break;
7.   end if
8. end for
9. end for
10. return \((e_1,l_1'),(e_2,l_2'),...,\(e_n,l_n)\);

---

### 5. Experiment Results

We manually labeled 246 destination names and 793 non-destination names from 100 documents. In order to avoid the problems caused by the imbalanced class distribution, we constructed a balanced data set by choosing all the 246 destinations and randomly selected 245 non-destinations as ground truth. We use 2-fold cross validation in our experiments. All numbers reported in this section are averaged results from the 2-fold cross validation.

Three machine-learning models are compared: Naive Bayes [7], Maximum Entropy (MaxEnt) [2] and C.4.5 Decision Tree [9]. The three models show similar performance. We show the results using the MaxEnt model. The tunable parameters are: \( d \) for threshold of domain registrant name matching; \( t \) for threshold for proportion of web page titles containing the named entity; \( p \) for phone book registrant name matching; \( c \) for threshold of cosine similarity in post-processing; and \( n_c \) for threshold of normalized count feature. Note that the scopes of \( y \)-axis in Figure 2 and 3 are from \(0 - 0.5\) and \(0.5 - 1\).

#### 5.1 Evaluation of External Features

We evaluate the effects of external features (Feature 6, 7 and 8) by comparing the performance of the models under 3 settings: (1) internal features only (Feature 1 - 5), no post-processing; (2) all features (Feature 1 - 8), no post-processing, and (3) all features plus post-processing. In Figure 2, these settings are represented by “Base”, “Base+Ext.” and “Base+Ext.+PPS.” respectively. Among the tunable parameters of external features, \( d \) has the largest impact on the performance. With the same \( d \), the effects of other parameters are similar. Therefore, in Figure 2, we varied \( d \) from 0.1 to 1.0, with \( step = 0.1 \).

The performance reaches the highest when \( d = 0.6 \). The external features reduced the total number of mis-classified named entities by 5.6%. After applying post-processing, the reduction reaches 16.9%, as shown in Figure 2(a). The overall accuracies increased from 81.8% (Base) to 82.9% (Base + Ext.) and then to 84.9% (Base + Ext. + PPS.). In terms of recall, 12.2% un-recognized destination names were picked up by external features, and 37.0% were picked up after applying post-processing. As shown in Figure 2(b), the precision dropped about 1% to 2%. With the two effects working together, the F1 score still increases from 80.4% to 81.9% (Base+Ext.) and then to 84.8%(Base+Ext.+PPS.).
Without the external features and the post-processing step, the overall classification accuracy is around 79%. External features and the post-processing step improved the accuracy to 84%.

5.2 Evaluation of Post-Processing

We trained and tested the MaxEnt classifier with all the features. We then applied the post-processing algorithm and gathered the overall accuracy, the precision, recall and F1 score of destinations. 10 values (from 0.1 to 1.0, step = 0.1) are used for all thresholds. All experiment results showed similar effects of the post-processing step. For simplicity and clarity, we show the results in Figure 3 under one setting: \(d = 0.5\), \(t = 0.1\), \(p = 0.3\), and \(nc = 0.04\).

Figure 3 shows that although post-processing decreases the precision but increases recall significantly. The overall effect is it increases the accuracy and F1 score. A threshold value of 0.7 gives us the best performance in this setting. It increases the overall accuracy and F1 score of destination by 2% and 3%, respectively.

5.3 Discussion on Feature Relevance

We calculate correlations and mutual information to capture the dependency between each feature set and the class label. The whole feature space is divided into 6 subsets: feature sets 1 - 5 are 5 subsets; features 6 - 8 together form one subset, called “external features”. Given a feature set, \(F\), and a named entity \(e\), a random variable \(x\) is 1 if \(e\) has a feature in \(F\), 0 otherwise. A random variable \(y\) is 1 if \(e\) is destination, 0 otherwise. Table 1 shows that external features have strong relevance with the destination.

6. REFERENCES