Evaluating Methods for Probabilistic Address Elementalization

Miria Grunick
New York University
mlg454@nyu.edu

Abstract

The identification of the individual components within an address, also known as address elementalization, is necessary for many types of applications including geocoding and address standardization. Traditionally, this work has been done with country-specific rule-based strategies. Since address components have a finite number of classes and are highly contextual, I propose that these components can be identified with a high accuracy by using part-of-speech tagging techniques with a custom grammar. In this paper, I examine the accuracy and speed of six different part-of-speech tagging techniques for address elementalization.

1 Introduction

Across international addresses, there are identical functions of the individual tokens. We will refer to the individual tokens as address components. These components can be names of administrative places, such as a country, state, county, city or neighborhood or they can be used to quantify a location on a street, such as the street number, prefix, pre-directional, street name, suffix or post-directional. Address elementalization is the process of determining the role of each individual term in a given address.

Unfortunately, address formats are not uniform across countries. Some countries, such as Ghana and Cameroon, do not use a standard postal code system. Even if postal codes are present, different countries may place the postal code at different positions in the address. Other countries, such as Austria and Poland, do not use state or province information in their address formats.

Historically, this problem has been solved by generating rule-based approaches tailored to country-specific address formats. Implementing new address parsers in rule-based systems is tedious as it requires a lot of knowledge of the country's language and address formats. Implementing systems that can identify global addresses is very time consuming as a separate processor must be tailored to each individual country.

There are some natural language processing techniques which are used for classifying items in a sequence where the components are contextually dependent on each other, such as part-of-speech tagging. Since addresses also have a contextual sequential order, it seems logical that these techniques would be well-suited for classifying the components of addresses.

2 Related work

Although a number of papers discussed address elementalization as a step in the data mining task of extracting addresses from text documents, there were only a few that focused on optimizing the process of address elementalization. Two papers that discussed this topic in depth are mentioned below.

Borkar et al. (2000) proposed a method that used two nested Hidden Markov Models (HMM) to elementalize addresses. The outer HMM was used to capture sequencing relationships between different element classes whereas the inner HMM captured the structure of each internal element class. In their paper, they parsed full American addresses into six address components (house number, box number, street, city, state and zip) and achieved a 99.6 % accuracy in correctly tagging the individual address components within these six classes.

Christen and Belacic (2005) also proposed a HMM-based solution for address elementalization. Their strategy used a HMM model and some lookup tables of known token states which were generated from both address format guidelines and

individual training addresses. During the classification phase, the addresses were first tagged with all possible known tag observations from the lookup tables as well as information about the word features, such as token length and the character types of the token. These tags were then used as the possible state observations for the model. The HMM was used to return the highest probability sequence over the generated observations. On a tag set of 26 different states, their model achieved a 97.40 % accuracy on full Australian addresses.

3 Grammar

As there is no standard grammar for address components, it was necessary to create one. The grammar I propose in Table 1 identifies twelve distinct address components.

Tag	Component
AA1	Country
AA2	State
AA3	County
AA4	City
AA5	Neighborhood
ZIP	Postal Code
NUM	Street Number
PFX	Prefix
PDR	Pre-directional
STR	Street Name
SDR	Post-directional
SFX	Suffix

Table 1: Proposed address component grammar

This grammar defines five different levels of administrative areas which correspond to five different levels of hierarchical place data. There is also a tag for postal codes. The rest of the grammar tags are related to components of the street address. Street names are the base names of the streets and street numbers indicate a position on a given street. Prefixes are any quantifiers that appear before the street name. For example, these are common in French-Canada and Mexico where the street type appears before the street name. Like prefixes, suffixes are street quantifiers that appear after the street name. These are more common in the United States and English speaking Canada where the street type frequently follows the street name. Pre-directionals and post-directionals are commonly cardinal directions that either precede or follow a street name.

4 Data

The source data used for evaluation is the proprietary TomTom North American Address Point data set. This data set contains both full address strings as well as columns containing the labeled address components. I chose to work with American, Canadian and Mexican address formats as they vary widely in both language and address formats. For this experiment, I selected approximately 450,000 address points per country as training sets and 250,000 address points per country as test sets. I also created other data sets with approximately 250,000 address points that could be used as either validation data sets or held-out training data sets. The selected points were randomly selected from the master data set and were shuffled.

I created two subsets of the address points to reflect two common cases of address elementalization. The first data set contained the fully qualified address as provided by the data. This format is commonly used when the full address is known but the address needs to be standardized. Each record contains all address components assembled in their correct order as determined by the country's address format. The second data set contained partial addresses. As users infrequently enter the entire qualified address when searching for a location, this is an approximation of how performance in the context of partial user input. The partial address data set contained generated partial formats such as street address and city, street address and postal code, city and state, state only, and postal code only.

5 Approach

I selected six common part-of-speech tagging strategies to evaluate for the task of address component tagging. For some of the Hidden Markov Model strategies, I also evaluated a few improvements to the unknown word models. I tested each strategy six times, using both full-format and partial address data for all three of the evaluation countries. All models were trained on test data of the same format. For example partial American addresses were both trained against and tested on a set of partial addresses. The final results were generated by running the trained model on the test data set.

5.1 Tagging Strategies

There were six part-of-speech tagging strategies that were evaluated. The first strategy (referred to as Most Frequent) was a simple strategy that did not rely on the context of a word to select a tag. For each word, it simply assigned the most frequently observed tag from the training set. If the word was not seen in the training set, it simply assigned the most commonly observed tag. The next two strategies were based on Hidden Markov Models (HMM). The first HMM strategy was a bigram HMM which used the previous state in calculating the transition probabilities between states. The second HMM strategy was a trigram HMM, where the previous two states were used in calculating the transition probabilities between states. The next strategy was a Maximum-Entropy Markov Model (MEMM). The MEMM was implemented with OpenNLP's Maxent maximum entropy library. I used the following features for generating the maximum entropy probabilities: the current term, the previous term, the next term, whether this term is the first term, whether this term is the last term, the term length, and the previous tag value. I also tried two other non-Markov Model implementations. The first strategy was a Transformation-Based Learning (TBL) tagger (also known as a Brill tagger) as implemented by JTBL (Black, 2011). The second strategy was a Conditional Random Field (CRF) tagger as implemented by LingPipe (Alias-i, 2008).

5.2 Unknown Word Models

Some models, such as Hidden Markov Models, perform poorly with unknown words. As many of the administrative and street names are not common words in a language, it was necessary to select a good unknown word model. I selected four common unknown word models to evaluate. The first model simply returned a very small fixed probability for any unseen word. The second model implemented a maximum entropy model to assign a probability. The maximum entropy model was implemented with OpenNLP's Maxent library. This model used the following features for generating unknown word probabilities: the current term, the previous term, the next term, whether this term is the first term, whether this term is the last term and the length of the term. The third model was a feature-based model which used multiple observed features of the word to assign a probability. The feature-based model used the following features in calculating the probability: the word length, whether the word was in all capital letters, whether the term is completely numeric and whether the word contains at least one number. The fourth model was a singleton model which modeled the unknown word probability after the probability of all words that were only seen once.

6 Results

6.1 Unknown Word Model

I evaluated the four unknown word models by implementing each model in a bigram HMM and evaluating the results of the test data. I looked at two metrics: the accuracy of correctly identifying an unknown word and the average classification time in milliseconds. I tested each unknown word model against both the full-format and partial American address data set. Tables 2 and 3 show the results from these tests.

Strategy	Unknown	Time
Fixed	90.76	0.089
Singleton	99.91	0.103
Feature-based	97.15	0.217
Maximum entropy	98.04	1.364

Table 2: Unknown word model results on full-format American addresses

Strategy	Unknown	Time
Fixed	63.67	0.050
Singleton	94.44	0.064
Feature-based	94.18	0.119
Maximum entropy	98.49	1.096

Table 3: Unknown word model results on partial American addresses

Looking at the unknown word model results, we can see that the singleton, feature-based and maximum had high accuracy over both test sets. The maximum entropy model performed the best on the partial address data set but was much slower than the other two models. For the remainder of this experiment, I used the singleton model as it had the highest overall accuracy on full-format addresses and was much faster than the the maximum entropy model or the feature-based model.

6.2 Evaluation Criteria

To evaluate the six strategies, I looked at three separate metrics: the overall accuracy of the strategy, the accuracy of correctly identifying unknown words and the average classification time in milliseconds. For the Hidden Markov Model-based strategies, I implemented the singleton unknown word model, as selected from the testing in Section 5.2.

6.3 Full-Format Address Results

I evaluated all six strategies with the full-format addresses. The results of these tests can be seen in Tables 4, 5 and 6.

Model	Overall	Unknown	Time
Most Frequent	88.29	72.03	0.003
Bigram HMM	98.37	99.91	0.109
Trigram HMM	98.64	99.93	4.117
MEMM	99.72	99.41	9.378
CRF	99.75	98.82	0.042
TBL	96.85	72.03	0.041

Table 4: American full-format address results

Model	Overall	Unknown	Time
Most Frequent	90.37	0.25	0.003
Bigram HMM	98.38	99.86	0.096
Trigram HMM	98.40	99.88	4.196
MEMM	99.70	99.67	8.750
CRF	99.81	99.91	0.041
TBL	99.89	99.07	0.063

Table 5: Canadian full-format address results

Model	Overall	Unknown	Time
Most Frequent	84.43	1.43	0.003
Bigram HMM	94.14	98.42	0.109
Trigram HMM	94.26	98.52	4.844
MEMM	99.26	97.35	8.963
CRF	99.58	96.66	0.039
TBL	97.32	71.43	0.058

Table 6: Mexican full-format address results

All contextual part-of-speech tagging techniques performed well against the full-format addresses. In general, the MEMM and CRF strategies were the top performers across the data sets. The TBL strategy had a slightly lower overall accuracy than the MEMM and CRF strategies, but it had a considerably worse unknown word accuracy. The HMM-based models had high accuracy for both American and Canadian addresses, but exhibited a considerably lower accuracy over Mexican addresses.

6.4 Partial Address Results

I evaluated all six strategies with the partial addresses. The results of these tests can be seen in Tables 7, 8 and 9.

Model	Overall	Unknown	Time
Most Frequent	74.80	65.30	0.002
Bigram HMM	75.17	94.44	0.061
Trigram HMM	79.17	95.32	1.801
MEMM	98.58	91.44	3.887
CRF	98.29	95.39	0.024
TBL	95.40	60.34	0.044

Table 7: American partial address results

Model	Overall	Unknown	Time
Most Frequent	88.11	0.34	0.001
Bigram HMM	87.52	98.10	0.059
Trigram HMM	91.35	98.43	1.791
MEMM	98.65	84.58	3.622
CRF	98.41	98.26	0.024
TBL	90.89	71.22	0.087

Table 8: Canadian partial address results

Model	Overall	Unknown	Time
Most Frequent	81.67	1.22	0.002
Bigram HMM	87.28	96.71	0.069
Trigram HMM	88.24	96.69	2.630
MEMM	98.37	85.59	3.747
CRF	96.97	98.44	0.024
TBL	95.75	65.30	0.050

Table 9: Mexican partial address results

In the partial addresses, there was a wider variance in the accuracy results across the data sets. The HMM strategies saw a significant accuracy drop from full-format to partial addresses, even though the unknown word accuracy remained high. The CRF, TBL and MEMM strategies saw only a slight accuracy decrease between two address formats, although the TBL strategy still suffered from a lower unknown word accuracy than the other two models.

7 Conclusion

Address elementalization is a hard problem due to the differences of address formats between countries. Although rule-based approaches can successfully accomplish this task, they require an in-depth understanding of the country's address format and language. Probabilistic approaches, namely part-of-speech tagging techniques, are well suited for this task due to the contextual nature of addresses. Even with incomplete addresses, many of the strategies yield high tagging accuracy. In particular, the Conditional Random Field strategy consistently tagged both full-format and partial addresses with an overall accuracy of 96 % or above across all three national address formats and could generally do so under 0.1 milliseconds.

Further work includes investigating additional part-of-speech tagging techniques such as memory-based tagging, support-vector machines and neural networks. Additional investigation into other options for improving the unknown state estimations could be done, including the use smoothing techniques such as interpolation and backoff models.

Acknowledgments

I would like to acknowledge TomTom for graciously providing me with a student evaluation license over their proprietary address point data set.

References

Alias-i. 2008. LingPipe (Version 4.1.0) [Software]. *Available from http://alias-i.com/lingpipe.*

Apache Software Foundation 2013. Commons Configuration. (Version 1.10) [Software]. *Available from http://commons.apache.org/configuration.*

Apache Software Foundation 2013. Commons IO. (Version 2.4) [Software]. *Available from http://commons.apache.org/io.*

Apache Software Foundation 2013. Commons Lang. (Version 2.6) [Software]. *Available from http://commons.apache.org/lang.*

Apache Software Foundation 2013. Commons Logging. (Version 1.1.1) [Software]. *Available from http://commons.apache.org/logging.*

Apache Software Foundation 2013. OpenNLP Maxent. (Version 3.0.3) [Software]. *Available from http://opennlp.apache.org.*

Black, Bill. 2011. Transformation-Based Learning in Java (Version 2011-03-17) [Software]. *Available from http://jtbl.sourceforge.net.*

Vinayak Borkar, Kaustubh Deshmukh and Sunita Sarawagi 2000. Automatically Extracting Structure from Free Text Addresses. *IEEE Data Engineering Bulletin*, 23(4):27-32.

Peter Christen and Daniel Belacic 2005. Automated Probabilistic Address Standardization and Verification. *4th Australasian Data Mining Conference AUSDM'05*