

# Named Entity Recognition for Housing Data

**Bang Liu**

Department of ECE  
University of Alberta  
Edmonton, AB, T6G 2E8, Canada  
bang3@ualberta.ca

## Abstract

We constructed a Named Entity Recognition (NER) system for our housing data. The dataset of Making Sense of Microposts (MSM) 2013 Concept Extraction Challenge was preprocessed to serve as our training dataset. Besides, we also annotated our housing data as the development dataset and test dataset. We tried different kinds of off-the-shelf tools, including Stanford NER, Twitter NLP and Apache OpenNLP. We also utilized the Edmonton Transport System (ETS) landmark list to find out location entities. Finally, the results of different methods were combined by training a classifier on them using Weka. The final system achieved great improvement compared with individual off-the-shelf tools and methods.

## 1 Introduction

Location is important to a house. In our housing data, every house was described by a paragraph of text. The sellers sometimes mentioned that their houses were close to some locations. If a location was mentioned many times, it is highly possible that this place is somewhere that people want to live close to. Finding out these locations is helpful to construct a model of house prices. To achieve this, we need an NER system for the housing data.

Most of the current NER methods have been designed for concept extraction from relatively long and grammatically correct texts, such as newswire texts or biomedical texts. However, the housing data is not long and often grammatically incorrect, which is more similar to the nature of microposts. The goal of our approach is to combine several different information extraction methods in order to reach a more precise result on our housing data.

In the next section, we describe different methods for NER task. Section 3 describes our task

and the challenges. Section 4 describes the construction of our NER system. Section 5 compares the results of individual methods and the merged result, and shows the results after adding post-processing. Section 6 talks about the conclusion and future work.

## 2 Paper Review

We referred to the MSM2013 Concept Extraction Challenge to figure out the main approaches taken by the state of the art systems (Basave et al., 2013). In this challenge, participants were required to classify four entity types: person (PER), organization (ORG), location (LOC) and miscellaneous (MISC).

From the results and participants experiments we made a number of observations. The best performing systems for the top 4, based on F1 score, were hybrid: they combined rule-based and machine learning methods. The success of these models appears to rely on the application of off-the-shelf systems, such as Stanford NER, OpenNLP, Illinois Wikifier and so on.

## 3 Task Description

Our task is finding out named entities of locations in the housing data. A sample of our housing data is as following:

“0031958960 Bedrock built 1860sqft, two story situated in Ellerslie Heights. Waiting on a new buyer this move in ready home provides main floor with rich hardwood floors. Open great room with built in sound system. Sizable living area with gas fireplace. Dining room with access to backyard. Kitchen with pantry, large island with raised eating bar and built in black appliances. Main floor laundry and double attached garage. Upper level offers a spacious

separate bonus room. Three bedrooms with a master suite that comes with a walk in closet, full luxury ensuite with stand-up shower, soaker corner tub and large vanity. The home also comes 90% fenced and with a large oversized deck. Located close to Anthony Henday, Calgary Trail, Nisku, South Edmonton Common, schools, park, trails and all major amenities. ”

The outputs are tokens of the text and their corresponding tags: “I-LOC” for location entities and “O” for other tokens. However, this task of identifying entities of location is hindered by:

- (i) the noisy lexical nature of the housing data, where terminology differs between users when referring to the same location;
- (ii) the text may not grammatically correct;
- (iii) many uppercased words are used to show the houses properties

As we can see, these properties of the housing data are kind of similar to the nature of microposts. Besides, we don’t have a tagged dataset from our housing data, so that we used the training dataset of the MSM2013 to train our taggers. We also annotated 300 paragraphs of housing descriptions: 150 of them were served as development dataset and the rest were test dataset. After tokenization, our development dataset contains 23,404 tokens, and 394 of them are I-LOC tokens. The test dataset contains 23,070 tokens and 386 of them are location entities.

## 4 Our Approach

Our approach incorporates several different well-known NER tools and methods: Stanford Named Entity Recognizer (Stanford NER) (Finkel et al., 2005), Twitter NLP (Ritter et al., 2011), Apache OpenNLP (Open, 2008), and the method of searching from the ETS landmark list (landmark, 2014). After getting the results of different methods, we treat the results of every token as a feature vector of it. In this way, every token was transformed into a nominal vector, and every dimension of the feature vector was the classification result of one method. An overview of our system is shown in Figure 1. In the remainder of this section, we explain the components in detail.

**Preprocessing:** In the preprocessing phase, we tokenized our development dataset and the test dataset, and tagged them. Every line contains a token and its tag, O or I-LOC, separated by a tab. The MSM2013 training dataset is also transform into the same format with the development and test dataset, and we retained the tags (I-PER, I-LOC, I-ORG, I-MISC and O).

**Stanford NER:** The Stanford NER system (version 3.3.1) is retrained on the MSM2013 training dataset, using parameters based on the properties file *english.conll.4class.distsim.prop* provided with the Stanford distribution. The Stanford results serve as a baseline, as well as input for the hybrid classifier. Besides, we directly used the *english.conll.4class.distsim.crf.ser.gz* classifier provided with the Stanford distribution to get another classification result.

**Twitter NLP:** The entities extractor of the Twitter NLP is well trained on millions of Twitters. So we didn’t retrain it and used it directly with two kinds of different parameters settings. The first setting was take advantage of Part-of-Speech (POS) tags and chunk tags, and the second setting was based on POS tags and event tags.

**Apache OpenNLP:** We applied the pre-compiled name finder and models (*en-ner-person.bin* and *en-ner-location.bin*) of Apache OpenNLP (version 1.5.0) to the development dataset and test dataset.

**ETS landmark list:** To improve the recall rate, we got a list of ETS landmarks from the ETS website. For every token, if it exists in the list, then we tag it as I-LOC. As the landmark names in the list are long and complete, it often contains some normal words such as “the”, “of”, “home” and so on. So we did some post processing to change the tags of these tokens back into O.

**WEKA:** After finished the above steps, every token has 6 tags from the four methods. We put the 6 dimensions result vectors together with the correct tags into a table (a CSV file). One table for the development dataset and another one for the test dataset. We experimented with several different algorithms and machine learning settings using WEKA-3.6.10. Finally, we choose the classifier that gave out the highest F1 score on the development dataset, and test it on the test

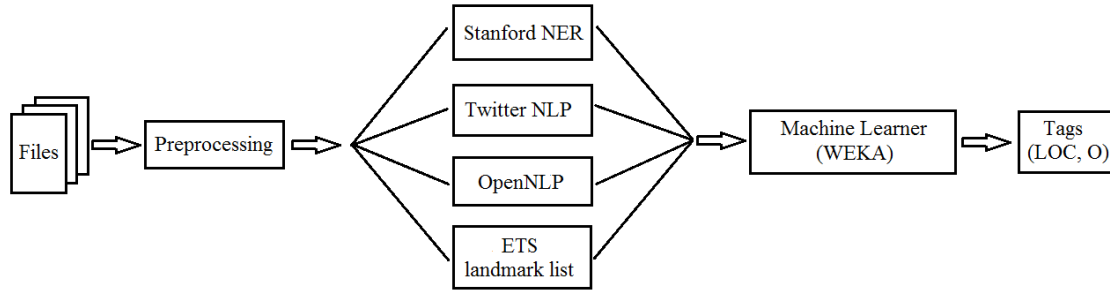


Figure 1: Proposed NER System Structure

dataset. This gave out the classification results of our NER system.

## 5 Experimental Results

We trained different classifiers on the development dataset using WEKA: Bayes classifiers such as AODE, AODEsr, BayesNet, NaiveBayes; functions such as LibSVM, Logistic, MultilayerPerceptron, RBFNetwork; trees methods such as RandomForest, RandomTree, and so on. Based on the F1 score on the development dataset, the best classifier was AODEsr (Zheng and Webb, 2006). Table 1 shows the different results of the development dataset given by individual methods and the combined system. Table 2 shows the results of the test dataset. Notice that the precisions, recalls and F1 scores are that of I-LOC tokens. The “Stanford 1” is the baseline experiment setting. Other experiment settings correspond to what we described above in sequence.

Table 1: Results on the development dataset

Method	Precision	Recall	F1
StanfordNER 1	0.3857	0.1371	0.2022
StanfordNER 2	0.1669	0.2995	0.2144
TwitterNLP 1	0.5723	0.2411	0.3393
TwitterNLP 2	0.4265	0.368	0.3951
OpenNLP	0.4967	0.1929	0.2779
ETS list	0.2152	<b>0.6193</b>	0.3194
Combination	<b>0.684</b>	0.609	<b>0.644</b>

In table 1 and table 2, we highlighted the best precision, recall and F1 score in bold. As we can see, our NER system outperformed individual methods by a big percentage. To make the results comparison more intuitive, we can look at figure 2 and figure 3. The highest precision rates and F1 scores were given by our system both on the devel-

Table 2: Results on the test dataset

Method	Precision	Recall	F1
StanfordNER 1	0.561	0.1788	0.2711
StanfordNER 2	0.2536	0.3187	0.2824
TwitterNLP 1	0.5692	0.1917	0.2868
TwitterNLP 2	0.4514	0.3005	0.3608
OpenNLP	0.5207	0.1632	0.2485
ETS list	0.2385	<b>0.6036</b>	0.3419
Combination	<b>0.78</b>	0.58	<b>0.666</b>

opment dataset and on the test dataset. The highest recall rates were given by the ETS landmark list method, which was not surprising. However, our system’s recall rates were quite close to the best.

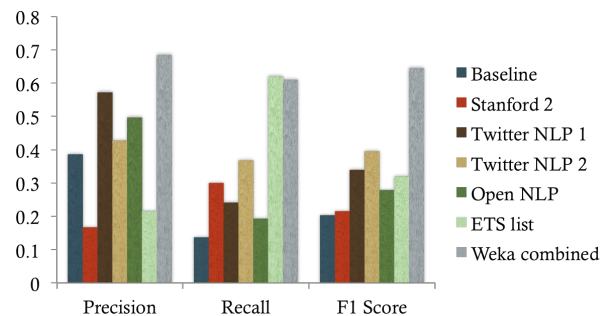


Figure 2: Comparison of results on the development dataset

There are several reasons why the hybrid NER system beats individual NER tools and methods. First, different tools are able to recognize different named entities due to their own properties. When they are combined together, they can be complementary to each other. This helps to improve the recall rate. Second, one can imagine that it is more plausible that a word is an entity when multiple services claim this with high confidence than when only one service claims this with low confidence.

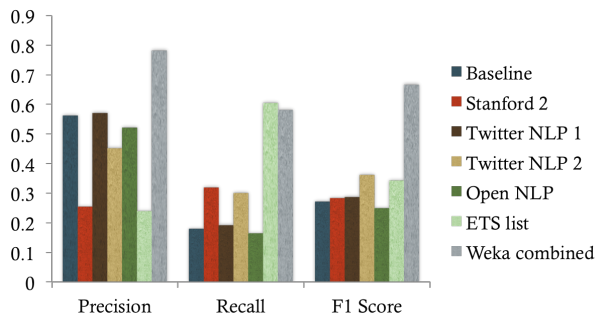


Figure 3: Comparison of results on the test dataset

That is why the precision rate improved a lot. With the improvement of both precision and recall rate, the hybrid NER system outperforms off-the-shelf approaches.

To improve our system, we looked into the incorrectly classified tokens. We discovered some frequent error patterns: street names such as “43 Street”, “161 Avenue”, “32 Ave” and so on; abbreviations such as “NW”, “SW”; some punctuations such as “,”, etc. There are also several other kinds of errors. To eliminate these disadvantages, we can try two strategies: the first is try to get more training dataset which contains more tokens that similar to what the NER system usually classified incorrectly, the second is construct a rule based filter to do some post-processing to the output of our hybrid NER system.

We chose the second method. We constructed a rule based filter to eliminate frequent error patterns, such as “43 Street”, “161 Ave” and so on, using regular expression. Table 3 and table 4 compares the results of our NER system with post-processing and the results without post-processing.

Table 3: Final results on the development dataset: adding post-processing

Method	Precision	Recall	F1
Combination	<b>0.684</b>	0.609	0.644
Post-process	0.6515	<b>0.6548</b>	<b>0.6532</b>

Table 4: Final results on the test dataset: adding post-processing

Method	Precision	Recall	F1
Combination	0.78	0.58	0.666
Post-process	<b>0.7982</b>	<b>0.6762</b>	<b>0.7321</b>

As we can see, the recall rate and F1 score improved after post-processing. By adding post-processing step to our NER system, the final results, include precisions, recalls and F1 scores, beat all individual off-the-shelf tools and methods that incorporated in our system.

## 6 Conclusion

Extracting named entities of locations from the housing data is a difficult task due to the un-formal nature of the data. In this project, we have shown that the combination of different NER systems and methods outperforms off-the-shelf approaches, as well as the customised Stanford approach.

Our future work contains the following plans:

1. Annotate our own training dataset from the housing data;
2. Try to incorporate more off-the-shelf NER systems;
3. Do more post-processing on the classification results based on the error patterns of the NER system.

## References

- Amparo Elizabeth Cano Basave, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. 2013. Making sense of microposts (#msm2013) concept extraction challenge. In *MSM*, volume 13, pages 1–15.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.
- ETS landmark. 2014. Website: <http://opennlp.sourceforge.net>.
- NLP Open. 2008. Website: <http://opennlp.sourceforge.net>.
- Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.
- Fei Zheng and Geoffrey I. Webb. 2006. Efficient lazy elimination for averaged-one dependence estimators. In *Proceedings of the Twenty-third International Conference on Machine Learning (ICML 2006)*, pages 1113–1120. ACM Press.