A System for Extracting Structural Information from Online Advertisements

Kevin Jue, Nam Wook Kim
Department of Computer Science
Stanford University
June 7, 2009

Abstract

We describe a machine learning system for extracting structural information from online advertisements. Although we narrow our problem domain to Craiglist's book advertisements, our system can be applied to other advertisements such as housing advertisement or job postings. The system identifies six different entities which include book title, author, edition, price, location and contact information. It makes use of extensive local features as well as Amazon web service and US census gazetteers as external resources.

Keywords: advertisement, named entity recognition, maximum entropy markov model, sequence classification, machine-learning.

1. Introduction

1.1. Motivation

Our work is motivated by the fact that advertisements posted in a public bulletin board such as Craiglist tend to be half-structured. The primary reason is that users are not required to post advertisements in strict structured format. Even with a search system, such heterogeneity of the web advertisements places cognition load for users to find information of their interest. By extracting structured information, we will be able to reduce users' search time as well as to improve search performance.

1.2 Problem Definition

We want to extract specific parts of book ads on craigslist ads to store in a structured location, such as a database. The following parts that we want to extract are the ads' title, author, price, location, edition, and seller contact information (email and/or phone number). Also, some ads contain multiple books on sale. These ads generally have titles that have generic titles, such as "Various Books". Our project also attempts to extract these generic titles.

Extracting information from craigslist presents a variety of challenges. The ads on craigslist are written by many different people, with no common structure. The ads in general don't contain well formed sentences. Also, a few categories that we do try to extract initially looks difficult. Extracting book titles is not very simple. The title can contain words that are very generic. For example, one of our training data's ad has a book title:

"How to Get the Healthcare You Want".

Another difficult problem is classifying generic titles. These titles are more difficult than regular titles in that they are even more generic. Also, generic titles have many of the same characteristics of regular titles, such as every word in the generic title having a capital first letter.

1.3. Related Works

We got some inspiration for our project from a previous cs224n course final project. The paper written by Nipun Bhatia, Rakshit Kumar, Shashank Senapaty [1] described their approach to extracting information from craigslist automobile ads. We found that automobie ads are somewhat more easier to extract than other advertisements. It is for the reason that named entities in such ads are single words, not multiple words. For example, named entities that they used include 'brand' and 'model' whose instances are 'Toyota', 'Hyundai' or 'Prius' etc. Therefore, their information extraction technique can hardly be applied to other advertisement domains such book advertisement. As explained in problem definition section, book titles and author names can be composed of arbitrary multiple words.

Most of our local features are inspired by Shipra Dingare et al [2]. They used a variety of features describing immediate context of each word, including words, n-grams, part of speech tags, previous lables, abbreviations, character substrings and word shapes. For word shapes, they take advantage of characteristics of biomedical texts. For example, many biomedical named entities include numbers, upper case letters and greek letters. We did not borrow these orthographic features, because of different problem domain. We also did not use abbreviations and character substrings for the same reason.

Our idea of using external features was also from Shipra Dingare et al [2]. Their motivation to use external resources is that local features sometimes were not able to provide sufficient evidence for confident recognition and classification. One of the major hurdles in our work was lack of clues to distinguish author names from book titles. We found that author names are frequently labeled as book titles. This paper gave us the inspiration to use amazon's web service as well as US census gazetteers to help with trying to classify authors.

2. System Components

2.1 Data Compilation

To retrieve the data from craiglist, we used their rss feed feature, instead of screen scraping. RSS is much better structured than html, so retrieving information this way was much less time consuming. The rss had a section for the title and one for the body of the advertisement. We extracted both of these sections and saved them into seperate files. The body still contained the html embedded in it. We decided not to remove the html tags since it initially looked like those tags can help with classification. We downloaded around 100 auto ads from the SF Bay Area craigslist site. We split the data into two sets, with a ratio of 80:20 for the training and testing set respectively.

2.2. Data Processing

After downloading the ads, we manually annotated the ads to identify all the labels. We used the Stanford University's NLP group's tagger (http://nlp.stanford.edu/software/tagger.shtml). We also tagged each word in the ads with their part of speech using python's nltk library (http://www.nltk.org/).

2.3 Classifier

We decided to use an MEMM classifier to extract all the book advertisment labels. We reused our cs224n programming assignment 3 MEMM classifier for this project. The original MEMM code was somewhat limited in a sense that we could not use part of speech tags, ngrams and external sources. We also could not use second or third previous labels. Thus, we did have to make some modifications to the classifier for our specific problem as well as for better performance.

We decided to train title and training data separately. The reason for this is that the structure of these two types of data is different, and the weights for all the learned features would probably be very different between the types of data.

2.4 Performance Evaluations

To determine how well our classifier is performing, we used the same evaluator used in programming assignment 3. This evaluator calculated the precision, recall, and f-score of all the phrases in the testing data. A phrase is marked as correctly labeled only if all the tokens in the phrase was correctly labeled and that none of the tokens that immediately precede and succeed the phrase is labeled as the phrase's label. Basically, the length of the guessed phrase must equal the length of the correct phrase.

3. Feature Set

3.1 Local Features

Selecting features is one of the most important aspects of named entity recognition problems. The main purpose is to find textual attributes that contribute to improving accuracy. It also depends on the context of the given problem, in our case book advertisements. In order to deal with special phenomena of advertisements in Craiglist, we derived a diverse set of local features which make use of the immediate content and context of each word.

We categorize our feature set into two subsets, orthographic features and semantic features. each of them is in turn divided into conjunctive and non-conjunctive features. The orthographic features include word shape such as starting with upper case letter, all letters capitalized and generalized word shape classes etc. Semantic features include vocabularies, part of speech tags and previous labels. In contrast to non-conjunctive features, conjunctive features are combination of non-conjunctive features. Analysis with these features will be discussed in Section 4. Feature Set Analysis along with performance results.

3.1.1. Default Features

Feature #	Feature	Example
1	Current Token	Word_i
2	Previous Label	NE_{i-1}

<Table 1> - For sequential classification, at least one of previous labels should be used in some of features. Otherwise, there would not be a sequence component to the

sequential model.

3.1.2 Non-Conjunctive Orthographic Features

Feature #	Feature	Example
3	Token Starting with Upper Case Letter	House, Used, Edition
4	All Letters in Token are Upper Case Letters	MAGAZINE, GEORGE
5	Uncondensed Word Shape	House->Xxxxx 5th -> dxx
	Condensed Word Shape	Shape$_i$: House -> Xx 5th -> dx
7	Previous Condensed Word Shape	Shape$_{i-1}$
8	Next Condensed Word Shape	Shape$_{i+1}$
9	Token enclosed by paranthesis	(glen park), (campbell)
10	Token Ending in 'st', 'nd', 'rd', 'th'	3rd, 7th, 1st, 2nd

<Table 2> - We have two types of word shape classes, condensed and uncondensed word shapes. For condensed shape class, we convert sequence of upper case letters to 'X', sequence of lower case letters to 'x', and sequence of digits to 'd'. For uncondensed shape class, we convert all upper case letters to 'X', lower case letters to 'x', and digits to 'd' without condensation.

3.1.3. Conjunctive Orthographic Features

Feature #	Feature	Example
11	Word Shape	Shape$_{i-1}$+Shape$_i$ Shape$_i$+Shape$_{i+1}$ Shape$_{i-1}$+Shape$_i$+Shape$_{i+1}$

<Table 3> - We only used condensed shape class, not uncondensed shape class, to make conjunctive orthographic features.

3.1.4. Non-Conjunctive Semantic Features

Feature #	Feature	Example

12	Token	Word$_{i-1}$ Word$_{i+1}$
13	POS	POS$_{i-1}$ POS$_{i}$ POS$_{i+1}$
14	Previous Label	NE$_{i-2}$ NE$_{i-3}$

<Table 4>- When running testing data, second and third previous labels are selected using best-scored labels of the first previous label during the viterbi decoding process.

3.1.5. Conjunctive Semantic Features

Feature #	Feature	Example
15	Token + POS	Word$_{i}$+POS$_{i}$ Word$_{i-1}$+POS$_{i}$ Word$_{i+1}$+POS$_{i}$

<Table 5>- One example of Word$_{i}$+POS$_{i}$ is 'The + DT' which is sometimes an indicative feature of starting point of a book title.

3.1.6 Tested, but Unused Local Features

Feature #	Feature	Example
17	Bigram	Word$_{i-1}$ + Word$_{i}$ Word$_{i}$ + Word$_{i+1}$
18	Trigram	Word$_{i-2}$+ Word$_{i-1}$ + Word$_{i}$ Word$_{i}$ + Word$_{i+1}$+Word$_{i+2}$
	Previous Label	NE$_{i-2}$+NE$_{i-1}$ NE$_{i-3}$+NE$_{i-2}$+NE$_{i-1}$
19	Token + Shape	Word$_{i-1}$+Shape$_{i}$ Word$_{i+1}$+Shape$_{i}$
20	Previous Label + Word	NE$_{i-1}$+Word$_{i}$
21	Previous Label + POS	NE$_{i-1}$+POS$_{i-1}$+POS$_{i}$ NE$_{i-2}$+NE$_{i-1}$+POS$_{i-2}$+POS$_{i-1}$+POS$_{i}$
22	Previous Label + Shape	NE$_{i-1}$+Shape$_{i}$ NE$_{i-1}$+Shape$_{i+1}$ NE$_{i-1}$+Shape$_{i-1}$+Shape$_{i}$
23	Previous Label +Shape + POS	NE$_{i-2}$+NE$_{i-1}$+POS$_{i-2}$+POS$_{i-1}$+POS$_{i}$

<Table 6> - All the above features are conjunctive semantic features. Because of the small size of training data, these features did not perform as expected. For example, most of bigrams and trigrams in test data have not been seen in training data, meaning that their

frequencies are mostly either zeros or ones.

3.2 External Features

We made use of a few external features. We tried these after using our local features, with this being more of a complementary part of this project.

The first external feature that we used is Amazon's product database. Amazon exposed all of their product information with a web service. We used this to try to help our classifier classify authors. To do this, we wrote a script that would retrieve the product listings for each word in our title and body data as well as each bigram. We ran this script for a few days and saved all of amazon's product data into a local database.

Another external feature that we tried is using last name and first name gazetteers from the U.S. census (http://www.census.gov/genealogy/names/names_files.html). We created a boolean feature that would return true if a given token is present in either of the first of last name lists.

4. Feature Set Analysis

Deciding on what features to use is an iterative process. We took the approach for first trying a very small set of features and analyzing the results to find out what our classifier got right and what it got wrong. Then based on what it got wrong, we added more features to our features set. We did our analysis on the title and body section of the ads separately.

4.1 Title Section Analysis

We initially thought that our title classifier would have a much better performance than our body classifier. The ad titles are much more structured than the ad bodies, and contain less noises. An example of a title is 'Data and Computer Communication by William Stallings (fremont) $10'. Most of ad titles are written in the similar format. The book title comes first. It is then followed by author, location and price in order.

4.1.1 Default Features

The first set of features that we tried is the default MEMM feature set that came with the programming assignment 3 MEMM model. The default feature set includes the token's previous label and the word itself. Running this, we got the following results:

Label	**Precision**	**Recall**	**F-Score**
Author	0%	0%	0
Edition	0%	0%	0
Generic Title	0%	0%	0
Location	91.67%	68.75%	78.57
Price	80%	85.71%	82.76
Title	29.41%	45.45%	35.71
Overall Score	58.33%	54.90%	56.57

<Table 7> - Title section data default features score

In general, the performance was poor for most of the categories. This is not unexpected, as the default features are very basic. This feature set did perform well on Location and Price, though. The tokens that were correctly labeled as LOCATION was because those tokens wer seen in the training data.

The reason why the PRICE label performed well is that all the phrases that are labeled as PRICE has the first token '$', and the second token as the price amount (note that we separated out the price strings into two tokens. For example the string '$5' was split into the tokens ('$', '5'). The two features in the default feature set were able to catch these two patterns.

In addition, using previous label as a feature contribute to correct classification for book titles. It is for the reason that book titles are composed of multiple words. Once the first word in a book title is correctly labeled, following words are likely to be classified to the label.

We also noticed that in general, this classifier will categorize a token as the same label as the previous token, and will only stop if it hits a token that it saw frequently in the training data. For example, the TITLE label continued until it hit the token "(glen park)", which was seen frequently in the training data as a LOCATION:

TOKEN	Wayne	Weiten	7th	Edition	psychology	(glen park)	$	100
CORRECT LABEL	AUTHOR	AUTHOR	EDITION	EDITION	TITLE	LOC	PRICE	PRICE
GUESSED LABEL	TITLE	TITLE	TITLE	TITLE	TITLE	LOC	PRICE	PRICE

<Table 8> - Title section Example 1

4.1.2 Non-Conjunctive Orthographic Features

With the addition of non-conjunctive orthographic features, our classifier got the following results:

Label	**Precision**	**Recall**	**F-Score**
Author	100%	40%	57.14
Edition	50%	50%	50
Generic Title	0%	0%	0
Location	100%	100%	100
Price	100%	100%	100
Title	50%	72.73%	59.26
Overall Score	70.69%	80.39%	75.23

<Table 9> - Title section data non-conjunctive orthographic results

There were significant improvements in most of categories including AUTHOR. We found that the reason for the improvement of the AUTHOR category was from two features, feature#5 (word shape classes) and feature#3 (starting with upper case letter). Feature#5 at first doesn't stand out to be able to help the AUTHOR category. But after some more analysis, we realized that the reason this feature helped is that most authors are preceded by the word 'by'. The general condensed shape of 'x' is actually quite infrequent in the training data, so our classifier assigned a high weight to that feature for the AUTHOR category.

At first glance, many of the added features should have greatly helped our EDITION category. We felt that feature#5 and #10(tokens ending in 'st','nd','rd' and 'th') should have been able to make our classifier correctly guess nearly all the EDITION phrases. So we were a bit surprised to see that we got a F-Score of only 50. After some analysis, we found that there were only two instances of EDITION phrases in our testing data, and the one we wrongly categorized was not because our features had a hard time labelling EDITION tokens. The table below shows the sentence where we mislabeled the EDITION phrase.

TOKEN	Wayne	Weiten	7th	Edition	psychology	(glen park)	$	100
CORRECT LABEL	AUTHOR	AUTHOR	EDITION	EDITION	TITLE	LOC	PRICE	PRICE
GUESSED LABEL	TITLE	TITLE	EDITION	EDITION	EDITION	LOC	PRICE	PRICE

<Table 10> - Title section Example 2

Our classifier labeled the token 'psychology' as EDITION mainly because if followed a token that was labeled EDITION. The 'psychology' token itself has no real distinguishing characters, so our classifier simply used the previous token feature. We believe that semantic features would improve this token's classification.

The LOCATION and PRICE categories performed very well. Those two categories have very distinguishing characteristics, and we had features that looked specifically for those characteristics such as feature#9 (enclosed by parentheses), so it was not surprising that we got an F-score of 100 for both of those categories.

The main reason why our TITLE category performance improved is not that our classifier was better in labeling TITLES per se, but that it was better in labelling other categories. With only our default features, our classifier had very low precision for the TITLE category, which affected both our recall and precision score. It mislabeled many tokens as TITLEs especially for tokens that came right after a correctly labeled TITLE token, which lowered our recall score.

4.1.3 Conjunctive Orthographic Features

We then added conjunctive orthographic features to our feature set. With those added features, our classifier got the following score on the title section test data:

Label	**Precision**	**Recall**	**F-Score**
Author	100%	60%	75.00
Edition	50%	50%	50.00
Generic Title	0%	0%	0
Location	100%	100%	100
Price	100%	100%	100
Title	52.94%	81.82%	64.29

Overall Score	71.67%	84.31%	77.48

<Table 11> - Title section data conjunctive orthographic results

The added features improved our recall score for AUTHOR. By looking at the weights assigned to the features during training, we found that the feature that takes the conjunction of previous token shape and the current token shape contributed most to the improvement. An example of the correct AUTHOR categorization for this classifier is below.

TOKEN	Brainfire	-	Campbell	Armstrong	-	PB
CORRECT LABEL	TITLE	O	AUTHOR	AUTHOR	O	O
GUESSED LABEL	TITLE	O	AUTHOR	AUTHOR	O	O
PREVIOUSLY GUESSED LABEL	TITLE	O	TITLE	TITLE	O	O

<Table 12> - Title section Example 3

We got the tokens "Campbell" and "Armstrong" correct in this version of the classifier, whereas we used to label those two tokens as 'TITLE' in previous versions. The conjunction of the shape '-' and 'Xx' is a feature for the token 'Campbell'. That feature is true for many Authors in the training data, and that is reflected by the high weight that feature was assigned for AUTHORs. Another important aspect of that feature is that for TITLEs, that feature does not occur in the training data. So our classifier assigned a very low negative weight to it for the TITLE category.

We also saw a slight increase in the TITLE category. But the main reason for that is the same as that of section 4.1.2. We improved in our accuracy for AUTHOR, which in turn made our TITLE labeling more precise.

4.1.4 Non-Conjunctive Semantic Features

We then added non-conjunctive semantic features to our feature set. With those added features, our classifier got the following score on the title section test data:

Label	**Precision**	**Recall**	**F-Score**
Author	100%	80%	88.89
Edition	100%	100%	100.00
Generic Title	0%	0%	0
Location	100%	100%	100
Price	100%	100%	100
Title	56.25%	81.82%	66.67
Overall Score	73.77%	88.24%	80.36

<Table 13> - Title section data non-conjunctive semantic results

The AUTHOR and EDITION classification was significantly improved from the additions of these features. The EDITION phrase that we correctly labeled in this version that we previously incorrectly labeled is mentioned in section 4.1.2. In this classifier version, we labeled the 'psychology' token to 'O', which we previously labeled as EDITION. The feature that had the greatest affect in the change of 'psychology' classification is the 'previous POS equal to NN' feature.

After some analysis, we realized that this is not really a common pattern of the data, but a result of the POS tagger that we used. That POS tagger labelled nearly all nouns as pronouns (NNP). It had a simple rule that labeled all nouns that started with a capital letter as a pronoun, unless it is the first word of a sentence. Since most of the nouns in the title data are capitalized, there were only a few tokens tagged as NN. Tokens that did come after a tagged-NN token is more likely to be tokens that didn't have much significance, which were annotated as 'O'. So the cause of this improvement was caused by quirk in our system.

The phase "philip kotler" was correctly labeled as an AUTHOR in this classifier version. In our previous versions, the phrase was labeled as 'O'. This improvement can be attributed by the fact that it comes after the word 'by' (feature#12). We did mention that our classifier does capture this pattern in section 4.1.2. non-conjunctive orthographic features. However, the correct classification of AUTHORs in section 4.1.2 was also caused from the pattern of most authors having their first letter capitalized.

4.1.5. Conjunctive Semantic Features

We found no improvement at all with adding the conjunctive semantic features to our feature set (see 3.1.5 and 3.1.6 for relevant features). We believe that the main reason is that we did not have much training data. It seems that semantic features in general require alot of training data to be effective. The features that we tried, such as bigram and trigram features, need alot of training data. In our case, the bigrams that were seen in both the training and testing data were captured by our other features, such as the bigram "$ 1"(note that in our system, the string '$1' is split into two different tokens).

4.1.6. External Features

Our last set of features is our external features. We only used these features to try to help with classifying author names. We initially believed that these external features in combination with the feature set in section 4.1.4 will perform very well. Both of our external features, Amazon database and name gazetteers, however, performed poorly. In particular, the classifier performed worse on the AUTHOR phrases than our 4.1.4 version. The reason for the lack of performance increase is that there are many non-AUTHOR tokens that also are names. It is not uncommon for a name to appear in a TITLE. Also, there are words that are names but have other meanings, such as "Ed", which can be used for edition.

We tried to isolate external features by excluding all other features except default features and see how the performance of sequence classification is improved for AUTHOR. We especially look at the author name 'Wayne Weiten' which all of our classifier versions incorrectly labeled as TITLE. The previous classifiers had a hard time to label it, since the author name came at the beginning of the sentence.

TOKEN	Wayne	Weiten	7th	Edition	psychology	(glen park)	$	100
CORRECT LABEL	AUTHOR	AUTHOR	EDITION	EDITION	TITLE	LOC	PRICE	PRICE
GUESSED LABEL	AUTHOR	AUTHOR	AUTHOR	AUTHOR	AUTHOR	LOC	PRICE	PRICE

<Table 14> - Title section Example 4

The use of Amazon database was able to label 'Wayne Weiten' correctly. Words following the author name, however, were incorrectly labeled as AUTHOR. this is because we are also using previous label as a feature. With careful consideration on combining external features with other local features, we believe that the overal performance will increase better than the result of section 4.1.4. For example, we could use the number of search hits in external resources and one of local features as a conjunctive feature.

4.2 Body Section Analysis

We initially believed that the body portion of our classification would do poorly, since the data is very unstructured, much more so than the title data.

4.2.1 Default features

The performance results for the default classifier for the body data is shown below.

Label	**Precision**	**Recall**	**F-Score**
Author	0%	0%	0
Edition	0%	0%	0
Email	0%	0%	0
Generic Title	0%	0%	0
Phone Number	0%	0%	0
Price	75.68%	63.64%	69.14
Title	40.00%	25.53%	31.17
Overall Score	59.70%	25.64%	35.87

<Table 15> - Body section data default features results

The only category that performed well is price. The reason for this is the same as that for the default classifier results of the title section data. Prices start with a distinctive token, '$', and the classifier will label the next token as a price with high probability.

We found that for the TITLE category, a very large weight was put on the "previous label = TITLE" feature. So for TITLE phrases, the difficulty in categorizing TITLE phrases is determining the first token and the token that immediately succeeds the last token in a TITLE phrase.

Determining the first token is much more difficult than the latter. There is no distinctive feature for the beginning of a TITLE phrase. Determining the latter is not too difficult, however. Usually the last token is succeeded by specific tokens such as 'by' or '.', or is succeeded by the EDITION phrase. The default features did decently well in this regard, since it uses the current word as a feature. We believe that semantic and orthographic features will help in identifying the EDITION phrases that come right after a title, as well as identifying the beginning of the book title using POS (e.g. The + DT).

4.2.2 Non Conjunctive Orthographic features

The results for the classifier with non conjunctive orthographic features combined with the default features is below:

Label	**Precision**	**Recall**	**F-Score**
Author	20%	3.12%	5.41
Edition	71.43%	58.82%	64.52
Email	100%	75.00%	85.71
Generic Title	0%	0%	0
Phone Number	75.00%	50.00%	60.00
Price	95.45%	95.45%	95.45
Title	38.71%	25.53%	30.77
Overall Score	70.30%	45.51%	55.25

<Table 16> - Body data non conjunctive orthographic features results

It is not surprising that the performance for the labels EDITION, EMAIL, and PHONE_NUMBER increased. Those categories all have a distinct pattern. For example, email addresses will be tokenized to three tokens to the form ('x', '@', y). It will be very easy for our classifier to recognize the parttern. Phone numbers also have a very common general shape of "d-d-d", and our classifier recognized it as well.

We did misclassify one EMAIL phrase. It is for the reason that most first tokens of EMAIL phrases have a general shape of xdx. The one that we misclassified doesn't. We believe that adding the semantic features would help for this specific instance, especially the feature that looks at the next token. Also, we shouldn't have split the email into three separate tokens. There were phrases where the '@' symbol was used for reasons other than an email. For example, we saw the phrase "contact me @ 415-353-6323" in the training data. If we did not split up the email, then it would have a much more distinctive shape. We could have also added a regular expression feature that searched for a '@' and a '.' in the string if we did not split up the email strings.

The reason why we did not get a better score for the PHONE_NUMBER label is that we mislabelled a number of PHONE_NUMBER phrases in the training data. The general shape of a phone number should have been good enough to correctly label all of the PHONE_NUMBER tokens in the test data. However, we missed a number of phone numbers when we were annotating the data, and they were labeled as 'O'.

4.2.2. Conjunctive Orthographic Features

Label	Precision	Recall	F-Score

Author	33.33%	3.12%	5.71
Edition	83.33%	58.82%	68.97
Email	100.00%	75.00%	85.71
Generic Title	0.00%	0.00%	0.00
Phone Number	75.00%	50.00%	60.00
Price	97.56%	90.91%	94.12
Title	45.16%	29.79%	35.90
Overall Score	75.53%	45.51%	56.80

<Table 17> - Body data conjunctive orthographic features results

Adding conjunctive orthographic features did increase our score slightly. Adding conjunctive features have better exploited surrounding context of each word such as '$' + price amount or '#th' + 'Edition'. The performance of book titles then have benefited from the correctely labeled entities.

4.2.3. Non-Conjunctive Semantic Features

Label	**Precision**	**Recall**	**F-Score**
Author	55.56%	15.62%	24.39
Edition	92.31%	70.59%	80.00
Email	100.00%	100.00%	100.00
Generic Title	0.00%	0.00%	0.00
Phone Number	100.00%	66.67%	80.00
Price	97.62%	93.18%	95.35
Title	65.62	44.68	53.16
Overall Score	83.65	55.77%	66.92

<Table 18> - Body data non-conjunctive semantic features results

Most of categories had some improvements from non-conjunctive semantic features except PRICE which have already had good performance. The main reason for the improvement in the EDITION category is feature#13 (POS). The EDITION phrases that we correctly labeled in this version are phrases were the word 'Edition' was misspelled. The POS tagger tagged these words as NN. As we mentioned before, the nltk POS tagger didn't tag many tokens as NN. The word 'Edition' was one of the few words tagged as NN, so the POS feature had a high weight for the EDITION phrases.

Many of the improvements in TITLE are because we were better in determining the first TITLE phrase tokens. The ones that we now consistently got correct are tokens that are preceded by a the token '>'. Accordingly, the semantic feature#14 (previous word = '>') had a high weight for TITLE. In the training data, many of the ads had a format like so, with embedded html tags like the following:

<p><font>Tauck Word Discovery-The West-2008

The reason why the AUTHOR label improved in performance was that we now correctly identified AUTHORs that came after the word 'by'. The overall performance was still poor in that, unlike the title section data, many of author names in the body section data do not follow such format.

4.2.4. Conjunctive Semantic Features

We then added Word+POS conjuctive features and got slight performance increase

Label	**Precision**	**Recall**	**F-Score**
Author	66.67%	18.75%	29.27
Edition	100.00%	70.59%	82.76
Email	100.00%	100.00%	100.00
Generic Title	0.00%	0.00%	0.00
Phone Number	100.00%	66.67%	80.00
Price	97.62%	93.18%	95.35
Title	67.42%	48.10%	54.54
Overall Score	85.22%	56.33%	67.54

<Table 19> - Body data conjunctive semantic features results

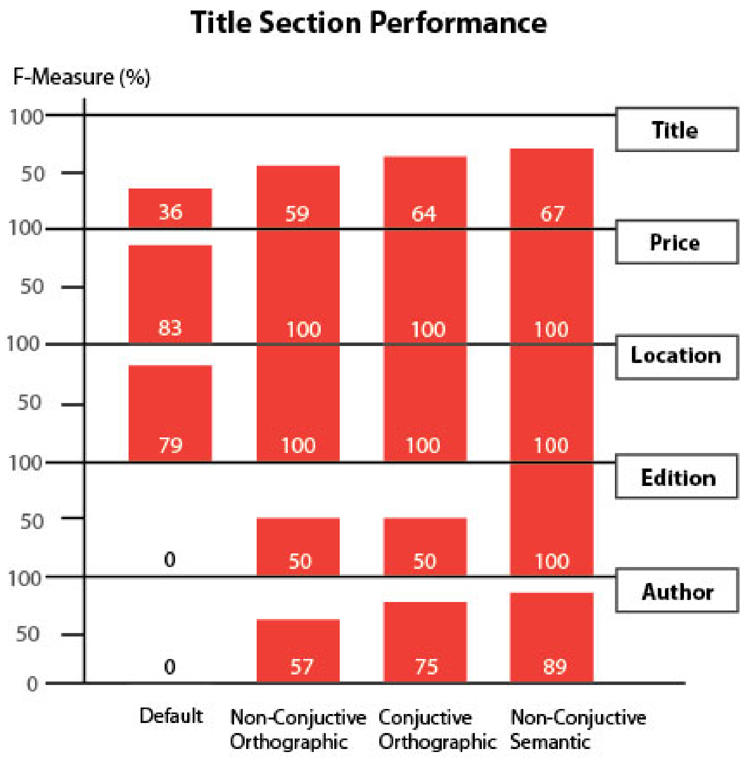
Although there was no significant improvement in performance, an interesting observation was that 'The'+DT sometimes correctly catch the beginning of the book titles.

4.2.5 External Features

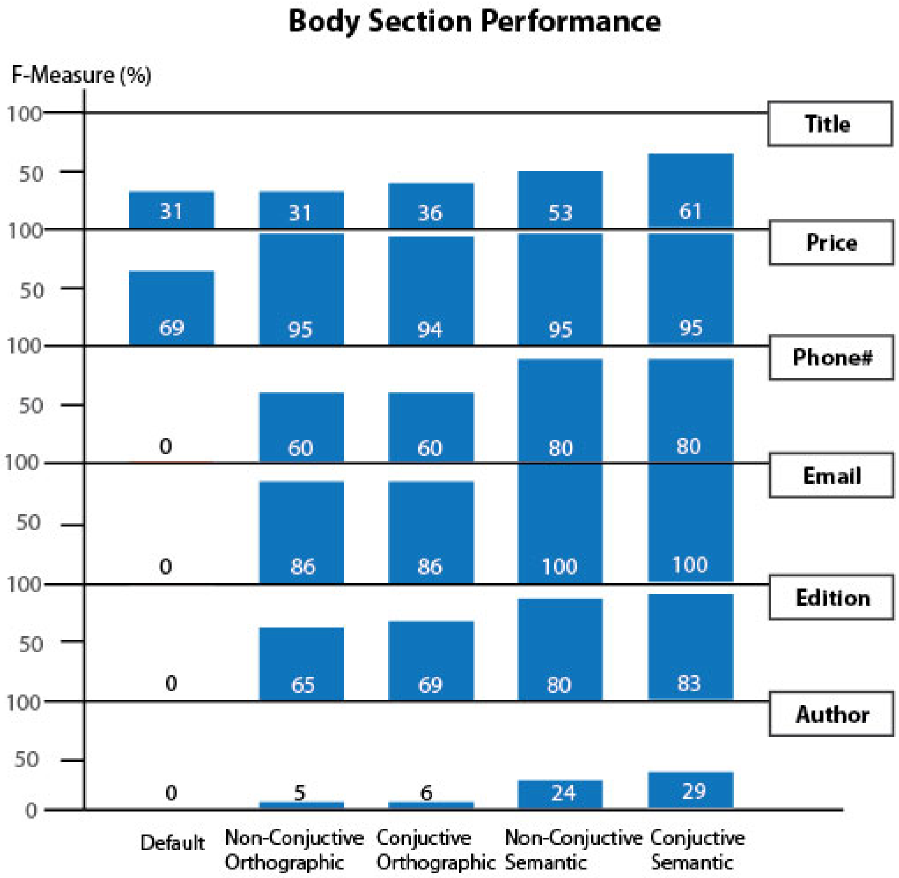
As with the title section data, external features did not perform well in the body section data. The reason is the same as that of the title data. There were many tokens that were names but were not classified as AUTHOR.

5. Conclusion

In this paper, we used MEMM for sequential classification task of book advertisements in Craiglist. Our system recognized book advertisement related named entities including book title and author name. We make use of a set of local features as well as external resouces. Even with challenges in advertisement corpus such as arbitrariness and broadness of book titles and author names, our system reached successful performance. The F-measures of our system performance are summarized in Figure 1 and 2. For title section, PRICE, LOCATION and EDITION are perfectly recognized as expected. TITLE and AUTHOR also achieved resonable scores, 67 and 89 respectively. For body section, PRICE, PHONE#, EMAIL and EDITION are mostly correctly labeled because of their well-formed structures in training data. Similar to title section data, TITLE got quite successful score (61), but AUTHOR got only 29. The AUTHOR was significantly affected by its previous label which is 'O' in most case.



<Figure 1> Performance result for title section data.



<Figure 2> - Performance result for body section data.

The respective contributions of semantic and orthographic features to performance was as expected. Considering author names and book titles, our initial hypothesis was that semantic features would work better than orthographic features. This is because of the fact that they are composed of multiple words and lack of special syntactic structure compared to location, price and contact information. Especially using previous label was the key to increased performance for book titles.

6. Future Work

An obvious future work would be using large training data. Most of our conjunctive semantic features did not perform well. We believe that such phenomenon was caused by the relatively small size of our training data. That is, the frequencies of the features were not enough to become effective. The deep semantic features with enough training data would also prevent author names to be labeled as book titles.

With limited time, we did not make better use of external features. We were not able to figure out best way to combine them with existing local features. For example, we have only tested adding the number of search hits in external resources with previous label as conjuctive feature. Finding the best combination is time consuming and difficult. In future work, we might adopt optimization method to automaticallly find best combination of different features.

In this paper, we were not able to improve accuracy for recognizing GENERIC TITLE at all. They were mostly recognized as TITLEs which seem resonable compromise to us. To better classify such category, we could employ additional features such as "containing plural lexicons like 'Books', 'Magazines' and 'Novels' etc". This will help us correctly label generic titles such as 'Used Books' and 'Various Magazines' etc.

The performance for AUTHOR category in body section data was very low. To improve this, we can add features specific to this category. Unlike the title section data, author names are not only preceded by 'by', but also separated by 'and' or special characters by '-' and ':'. For some reason, existing local features were not able to capture this pattern. The main reason was because previous label was so strong to gain most of weights. Adding those special features will evenly divide the feature weights in favor of AUTHOR category.

Acknowledgments

We would like to thank Professor Manning for his help helping define the scope of our project and guiding us in the right direction in this project. We would also like to thank Nate Chambers for giving us ideas on features to use as well as giving us feedback on our progress.

References

[1] Nipun Bhatia, Rakshit Kumar, Shashank Senapaty, "Extraction of Structured Information From Online Automobile Advertisements", http://nlp.stanford.edu/courses/cs224n/2008/reports/14.pdf

[2] Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, Claire Grover, "A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations".