Putting Semantic Information Extraction on the Map: Noisy Label Models for Fact Extraction

Chris Pal, Gideon Mann and Richard Minerich
Department of Computer Science
University of Massachusetts Amherst
Amherst MA, USA 01002

Abstract

Geographic indexing is a powerful and effective way to organize information on the web, but the use of standardized location tags is not widespread. Therefore, there is considerable interest in using machine learning approaches to automatically obtain semantic associations involving geographic locations from processing unstructured natural language text. While it is often impractical or expensive to obtain training labels, there are often ways to obtain noisy labels. We present a novel discriminative approach using a hidden variable model suitable for learning with noisy labels and apply it to extracting location relationships from natural language. We examine the problem of associating events with locations, where simple keyword matching produces a small number of positive examples within many false positives. Compared to a state-of-the-art baseline, our method doubles the precision of extracting semantic information while maintaining the same recall.

Introduction

Location-based indexing is a powerful way to organize information and a variety of compelling systems have been generating considerable recent attention (Toyama *et al.* 2003; Google Earth 2006; Google Maps 2006; Wikimapia 2006; Flickr 2006). Many of these systems rely on hand annotation or geo-tagging of information and media. However, there is a tremendous amount of information available on the web for which semantic association with locations could be obtained through natural language processing. We are interested in automatically deriving these types of semantic relationships to enable geo-spatial search and the display of results within compelling user interfaces such as (Google Earth 2006). In order to solve these problems, we turn to natural language processing methods, in particular *semantic information extraction*. Semantic information extraction is the task of identifying relationships of interest between entities mentioned in unstructured text.

Figure 1 illustrates two markers on a 3D Atlas for points of interest when visiting a town: (top) the birth location of Emily Dickenson – automatically identified from the text of a Wikipedia entry and (bottom) a local farmer's market – localized from the caption of images within a Blog. While our approach should be applicable to a wide variety of fact extraction tasks, we focus here on extracting location associations for events. Importantly, we are interested in methods with high precision as incorrect associations with locations would add significant noise into a search and browsing system.

We use this task to illustrate our contribution, a novel discriminative, hidden variable method for fact extraction that allows noisy data to be used for training. Our approach allows label noise to be explicitly modeled, effectively identifying false positives during learning. Our results indicate that this method can *double precision* for fact extraction while maintaining the same recall when compared with analogous models without hidden variables and without a label noise model.

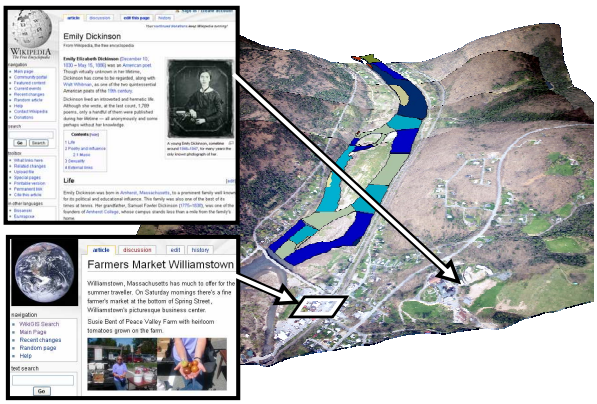

Figure 1: An example of a 3D Geospatial Interface to Wiki content for the birth place of Emily Dickenson as well as a entry for a local farmers market. Our algorithms automate the association of text and image content found on the web with map based interfaces.

Probability Models & Extraction

Machine learning techniques have proven to be powerful and effective for automating the construction of Internet Portals (McCallum *et al.* 2000). Furthermore, probabilistic machine

learning techniques are particularly attractive as they allow uncertainty to be treated in a formal and principled way. In this paper, we are concerned with semantic information extraction, where we are interested in obtaining precise relationships between entities such as images or events and locations. Semantic information extraction from the Web has had a long history, including (Brin 1998) who proposed an early model for building fact extraction systems using pattern matching. In recent years, general probabilistic models have been proposed for fact extraction. These methods allow larger and more flexible feature sets (Mann & Yarowsky 2005).

We model our problem in the following way: given a sentence s and a candidate relation r, define a set of feature functions $F = \{f_1, \ldots, f_n\}$. We then construct a classification model to predict whether the relation of interest is truly asserted in the sentence. This decision can be encoded as the binary random variable $y^{(s,r)}$. Consider first a naively structured random field for a collection of binary random variables for features. If we take each feature function to evaluate to a binary value when applied to random variable $x^{(s,r)}$ associated with that feature, we can write the joint distribution of labels $y^{(s,r)}$ and inputs $x^{(s,r)}$ as

$$p(y^{(s,r)}, x^{(s,r)}) = \frac{\exp(\sum_k \theta_k f_k(x_k^{(s,r)}, y^{(s,r)}))}{\sum_{x',y'} \exp(\sum_k \theta_k f_k(x'^{(s,r)}_k, y'^{(s,r)}))}. \quad (1)$$

Such models can also be described by naively structured factor graphs (McCallum *et al.* 2006; Kschischang & Loeliger 2001) as illustrated in figure 2 (Left). The various variants of both so-called naïve Bayes models and maximum entropy models, commonly used the the text processing community, can be illustrated using similar naïve graphical structures. However, there are a number of important differences. First, naïve Bayes models represent *joint distributions* as the product of an unconditional prior distribution on classes and class conditional distributions, typically discrete distributions for words or binary distributions for features

$$p(x^{(s,r)}, y^{(s,r)}) = \prod_k p(x_k^{(s,r)}|y^{(s,r)}) p(y^{(s,r)}). \quad (2)$$

When naïve Bayes models are used for words encoded as draws from a discrete distribution it is also possible to account for exchangeability. To fit such models, the Maximum Likelihood Estimate (MLE) of the parameters given training data $D = \langle d^{(1)} = \{x, y\}^1, \ldots, d^{(n)} = \{x, y\}^n \rangle$ can be computed by counting or equivalently, by computing sufficient statistics.

Conditional maximum entropy, or multinomial logistic regression models can also be illustrated using naively structured graphs. However, in contrast with naïve Bayes, such models are defined and optimized explicitly for the conditional distribution

$$p(y^{(s,r)}|x^{(s,r)}) = \frac{\exp(\sum_k \theta_k f_k(x_k^{(s,r)}, y^{(s,r)}))}{\sum_{y'} \exp(\sum_k \theta_k f_k(x_k^{(s,r)}, y'^{(s,r)}))}. \quad (3)$$

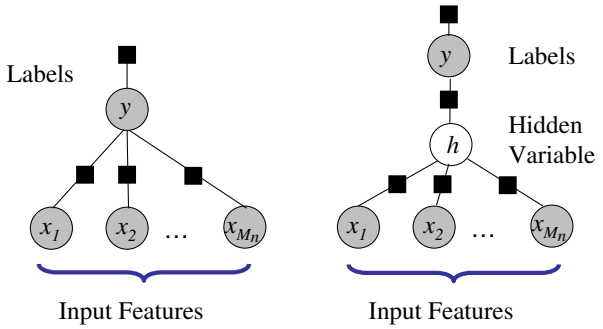

Figure 2: (Left) A Naive log-linear model as a factor graph. y is the noisy training label, and $x_{1..M_n}$ are the features. (Right) A hidden variable h representing the true label has been added to the naïve log-linear model.

The parameters of these conditional models are found by maximizing the log conditional likelihood of the training data

$$\hat{\theta} = \underset{\theta}{argmax}\ \ell(\theta; D)$$
$$= \underset{\theta}{argmax} \sum_d \ln p(y^{(d)}|x^{(d)})).$$

The optimization of parameters for these models can be performed with a variety of techniques including iterative scaling or gradient descent (Malouf. 2002). We use gradient based optimization and therefore use

$$\frac{\partial}{\partial \theta_k} \propto \sum_d \frac{1}{N} f_k(\tilde{x}_k^{(d)}, \tilde{y}^{(d)}) - \sum_d \sum_y p(y|\tilde{x}^{(d)}) f_k(\tilde{x}_k^{(d)}, y),$$

where $\tilde{x}$ denotes the observed value of variable x. A Gaussian prior on parameters is typically also used to help avoid over-fitting.

A Hidden Variable Model for Noise Reduction

Often, instead of a human annotating completely accurate labels $y^{(s,r)}$, it is quicker to create noisy labels $\hat{y}^{(s,r)}$, where these noisy labels are closely correlated with the correct human assigned labels but may contain errors. While this labeling often allows a dramatic reduction in the time needed to label examples, using noisy labels may result in lower performance than the comparative correct labeling. In order to reduce the errors from noisy labeling, we introduce an intermediate hidden binary random variable h with values corresponding to the true label assignment. We thus integrate over this hidden true label to obtain

$$p(\hat{y}|x) = \sum_h p(\hat{y}, h|x)$$
$$= \frac{\sum_h \exp(\sum_j \theta_j f_j(\hat{y}, h)) \exp(\sum_k \theta_k f_k(x_k, h))}{\sum_{\hat{y}',h} \exp(\sum_j \theta_j f_j(\hat{y}', h)) \exp(\sum_k \theta_k f_k(x_k, h))}.$$

Figure 2 depicts the difference between our models with and without the hidden label. The model is trained using the

noisy input $\hat{y}$, and in training it can, in essence, choose to "relabel" examples. In this way, the model can correct the errors from the noisy labeling during training by assigning what it believes to be the correct label h. This process can be seen as a form of semi-supervised clustering, where the true negatives and false positives are clustered together as are the true positives and false negatives. When we use this model for extraction, we thus integrate out the variable for the noisy label and use the prediction for the hidden variable h.

It is important to note that $\exp(\sum_j \theta_j f_j(\hat{y}, h))$ is a potential function that is constant across all examples, and encodes the noise model. For example, the potential encodes the compatibility that an example whose value is $\hat{y} = 1$ corresponds to the true label $h = 1$ (Table 1).

	true (hidden) label	
	h = 0	h = 1
y = 0	*true negative*	*false negatives*
y= 1	*false positives*	*true positives*

Table 1: Each table entry corresponds to feature functions $f_j(\hat{y}, h)$ for the hidden variable-label potential which encodes the noise model, e.g. the ratio of false positives to true positives. Later, we shall give example values for θ_js corresponding to each entry in the table.

As the experimental results below demonstrate, this can be a very effective method for noise reduction when the negative and positive examples are cleanly separated. In this case, the model will be able, in training, to identify examples which have been incorrectly labeled, correct these labels, and train a more precise model.

Parameter Estimation for Hidden Variable Models

Training models with hidden variables is more complicated than training a fully supervised model. (Salakhutdinov, Roweis, & Ghahramani 2003) propose the use of an expected gradient method, where:

$$\begin{aligned}\nabla \ell(\theta; D) &= \sum_d \frac{\partial}{\partial \theta} \ln p(y^{(d)}|x^{(d)}) \\ &= \sum_d \sum_h p(h|y^{(d)}, x^{(d)}) \frac{\partial}{\partial \theta} \ln p(y^{(d)}, h|x^{(d)}) \\ &= \sum_d \sum_h p(h|y^{(d)}, x^{(d)}) F(x^{(d)}, y^{(d)}, h) \\ &\quad - \sum_d \sum_{h,y} p(h, y|x^{(d)}) F(x^{(d)}, y, h),\end{aligned}$$

where F is the vector of all features. In the final form, the first term corresponds to the model's feature expectation over the hidden variable h given the observed label y, and the second term is the model's expectation over the hidden variables h and y. While models of this form are convex in the parameters, with hidden labels they become non-convex and optimization isn't guaranteed to find the global optimum on each run.

Event Location from Web Text

For many desirable fact extraction tasks, exhaustive annotation is often unavailable, for example for the "born-in" relationship. One alternative mode of generating labeled data is to write down example relations (e.g. born-in("Andy Warhol", "Pittsburgh")), which consist of a subject ("Andy Warhol") and a target ("Pittsburgh"). Next, two types of sentences are selected: sentences which contain both the subject and target and sentences which contain the subject and any other location. The former become positive training instances, and the later become negative training instances.

We automatically identify locations by named entity recognition. Named entity recognition is a well known technique in the Natural Language Processing (NLP) community for identifying members of open classes of nouns, such as people, organizations, or locations. In this paper, we use the OpenNLP toolkit (Baldridge, Morton, & Bierner 2002), which is an open source system for named-entity tagging that relies on a sequence level maximum entropy based classification model.

Given this set up, it is straight-forward to build a classifier $p(\hat{y}|x)$ to predict whether a given sentence x contains the relation or not. For a new sentence, with a subject and target indicated, you could then use the classifier to predict whether or not that pair has the relation of interest. This method works reasonably well.

However, this is a noisy method for collecting data. In particular, false positives are common and undetected. For example, there might be many sentences about "Andy Warhol" which also contain "Pittsburgh", but which don't say that he was born there (e.g. for the dedication of the Andy Warhol museum in Pittsburgh). Manual annotation of which sentences actually contain the relation of interest is prohibitively time consuming, and so previous training methods have simply ignored false positives in training and relied on future stages to compensate for low precision. However, this type of training has two key properties.

- The *false positives* closely resemble the *true negatives*. Mentions of a target by chance with a subject should appear to be mostly like mentions of anything of that type with the subject.[1]
- There are very few *false negatives*. The violations here will come from times when the database is deficient or the text is wrong.

To directly address training with false positives, we use the hidden variable model proposed above. In this context, the model admits an appealing generative interpretation: first we decide whether the sentence contains the desired relation and then we decide whether or not the relationship is true without regards for that particular sentence.

Given a set of labeled data $D = \langle d^{(1)}..d^{(n)} \rangle$, where each instance d is marked with a label $\hat{y}^{(d)}$ and a set of features $x^{(d)}$, we can then learn the model $\sum_h p(\hat{y}|h)p(h|x)$. This

[1]This assumption may be violated in certain cases, where for example, someone is more likely to be buried where they were born.

model should have a sharper distribution over **true positives**, $p(\hat{y} = 1|h = 1)p(h = 1|x)$, then the simple model would for $p(\hat{y} = 1|x)$, since it can separately model **false positives**, $p(\hat{y} = 1|h = 0)p(h = 0|x)$. Ideally, the learned distribution over h will yield a "clustering" on the inputs x, guided by the noisy labels $\hat{y}$. These clusters should exactly discover that $\{h = 0\}$ when the relationship doesn't occur and $\{h = 1\}$ when the relationship does occur, since $\{h = 0\}$ cases will resemble each other, no matter what the value of $\hat{y}$ is.

This model could be trained to estimate the label-hidden variable potentials.Alternatively, given the properties discussed above, we could construct a probability table expressing our relative confidences about possible outcomes and give it directly to the model, and have the model only learn $p(h|x)$. An example distribution for $p(\hat{y}|h)$ is shown in Table 2

$p(\hat{y}\|h)$	h = 0	h = 1
$\hat{y}$ = 0	.99	.01
$\hat{y}$ = 1	.6	.4

Table 2: Noise Model for Fact Extraction Training

This noise model encodes the notion that false negatives are relatively uncommon $p(\hat{y} = 0|h = 1) = .01$, while false positives are relatively common $p(\hat{y} = 1|h = 0) = .6$, in fact false positives are more common than true positives. We convert this noise distribution into a corresponding unnormalized hidden variable-label potential and hold it fixed in the model for the following experiments.

Experimental Results

In order to evaluate the above model, we selected the relation "born-in", and found a database of these facts on line for a set of celebrities. We then issued a query to Google for the celebrity's name, and downloaded the top 150 web pages for these celebrities. We then applied the named-entity recognizer described above and selected sentences which contained the celebrity's name and a location. For each location, we created a separate data instance, and marked it with $\hat{y} = \{0, 1\}$, if it exactly matched the key given in the database. This constituted a noisy labeling of all of the sentences.

For each data instance, we generated a set of features:

- The words in between the subject of the caption and the candidate location.
- A window of 1 around the subject and location.
- The numbers of words between the subject of the caption and the location.
- Whether or not another location appears interspersed between the subject and the location.
- If the subject and location are less than 4 words apart, the exact sequence of words between the subject and location.
- The word prior to the target.

	NB	MaxEnt	GModel-1	GModel-2
Accuracy	.944	.930 (.005)	.937 (.004)	.944 (.005)
Precision	.500	.254 (.032)	.337 (.046)	.503 (.144)
Recall	.085	.272 (.028)	.297 (.072)	.291 (.045)
F1	.145	.260 (.011)	.311 (.053)	.365 (.070)

Table 3: The hidden variable model with fixed label-hidden potentials (GModel-2) has double the precision of the MaxEnt model, demonstrating a significant noise reduction.

We then sampled some of these sentences and assigned labels $h = \{0, 1\}$, indicating whether or not the sentence actually contained the relation of interest (e.g. "born") or not. We used a strict decision method, only marking $h = 1$ when the sentence unambiguously stated that the person in question was born in the marked location. These were used only for evaluation and are the goal of discovery for the model.

Next we applied a naïve Bayes model, a maximum entropy model $p(\hat{y}|x)$ and the hidden variable model $\sum_h p(\hat{y}, h|x)$ to these sentences. We evaluated the system with regards to precision, recall, and F1 on the cases where $\{h = 1\}$. Since the goal is to use these extracted locations for augmenting a geo-spatial interface, the only relevant entities are the cases where the sentence actually mentions the location fact of interest.

Table 3 summarizes our Accuracy, Precision, Recall and F1 measures for extracting facts using a naïve Bayes model, a multinomial logistic regression model (MaxEnt), a hidden variable model with free label-hidden variable potentials (GModel-1) and fixed label-hidden variable potentials (GModel-2). While the naïve Bayes model has a high accuracy, its performance on the desired relations is the worst among the classifiers. GModel-1 is able to make some improvements over the maximum entropy model, the prior knowledge of the label-hidden variable potentials used for GModel-2 clearly helps. GModel-2 easily beats the maximum entropy model trained without a hidden label, and in the way that was expected, improved precision. This suggests that the model is able to pick out the false positives and model the true positives with a sharper distribution.

When comparing MaxEnt with GModel-2 we observe that precision is doubled, .254 for MaxEnt and .503 for GModel-2. While this level of precision may be low for direct use, the results of this type of extraction step are typically used as the input to a subsequent fusion step. For example, since we know that people only have one birth location we can pick the most confident location using a variety of means (Mann & Yarowsky 2005). More sophisticated scenarios can be thought of as re-ranking extracted facts based on their consistency with a probabilistic database. Both of these approaches would directly improve precision.

Table 4 and Table 5 show the highest weighted features for the associated hidden variable classes. The true positive class clearly has some very good word features ("born, birthplace"), and the false positive and true negative class also has some very good features ("nude"). Along with these good features are some odd features (e.g. "Theater"): this is a consequence of noisy web data.

Feature h=1	Value	w
BEFORE	,	2.5
BEFORE	Theatre	2.4
INTER	February	2.3
BEFORE	:	2.3
INTER	View	2.1
INTER	NY	2.0
INTER	Born	1.9
INTER	Birthplace	1.9
INTER	2005	1.7
INTER	1949	1.7
INTER	born	1.6

Table 4: The highest weight, w features for the cluster for hidden variable state 1. INTER features are words between the subject and target. BEFORE features come before the subject. DIST-1 indicates that the words are right next to each other. INTC indicates that another phrase of the target type is between the subject and target, while NO_INTC indicates the opposite.

Feature h=0	Value	w
INTER	Billy	2.8
INTER	$	2.6
INTER	.	2.0
INTER	Angelas	1.9
INTER	-	1.8
INTER	Los	1.7
INTER	US	1.7
NO_INTC	location	1.4
INTER	Nude	1.5
INTC	to	1.36
DIST-1		1.3

Table 5: The highest weight, w features for the cluster with hidden variable state 0.

Integrating Facts with Maps

The techniques we have presented here enable our final goal of associating a large number of facts extracted from the web with a map based interface. However, there are a number of ambiguities that may remain when processing natural language and extracting place names. For example, if the term "Springfield" is given in an annotation it is difficult to know if this refers to Springfield MA or NY, or OH, etc.

Accordingly, we have constructed a database of geo-referenced Wikipedia content through semi-automated information extraction techniques. We have also integrated a database of place names and GPS coordinates for locations in the United States of America. Using this information we can automatically associate plain text names with geographic locations. Subsequently, when names are mentioned in text we can leverage this information to automatically associated unstructured text annotations with numerical GPS coordinates. It is then possible to leverage our database of geographic locations and their attributes with statistical techniques such as those proposed in (Smith & Mann 2003) to resolve further ambiguity.

Related Work

Supervised semantic information extraction has been explored for a long time. (Chieu & Ng 2002) presents a maximum entropy model similar to what has been presented here for semantic information extraction. Unlike the models here, the model is trained on fully supervised data which has been manually annotated as to whether the sentence contains the relation or not, and does not have to contend with false positives in training. Given fully supervised training, these models can achieve high performance. However, fully supervised training is unlikely for the vast numbers of different types of events and relations of interest to potential users, and semi-supervised methods appear to be a crucial step in brining semantic information extraction to the masses.

The most closely related work is prior work in minimally supervised fact extraction. Models like (Agichtein & Gravano 2000; Ravichandran & Hovy 2002) use ensembles of weak classifiers, where each classifier has one feature which corresponds to a phrase. (Mann & Yarowsky 2005) demonstrated that these weak classifier models have lower recall and precision than the baseline methods presented in this paper (naïve Bayes and maximum entropy models). (Etzioni *et al.* 2004) proposes an alternative source of minimal supervision which contains instead of an example relationship, an example pattern which can extract that relationship, with some minor changes, the model presented here could be used in this minimal supervision context as well.

Somewhat more distantly related, (Hasegawa, Sekine, & Grishman 2004) presents early work on unsupervised semantic information extraction. The methods are typically more transductive than inductive, opertaing as unsupervised clustering as opposed to unsupervised classification.

Finally, (Lawrence & Scholkopf 2001) explores a related model for handling noisy training labels. There are a number of differences between his model and the one presented here, perhaps the greatest of them are the model he presents is a generative model and is applied to modeling gaussian process noise, as opposed to textual data. It is the analogue of the naïve Bayes model discussed above, which performs significantly worse than the maximum entropy model.

Conclusion and Discussion

This paper presents a novel discriminative hidden variable model. The model uses the given label as noisy training data, and learns a discriminative classifier for a hidden variable. In evaluation, the model estimates the hidden variable exclusively in order to classify new data instances.

We evaluate the model in the context of geospatial fact extraction, where the goal is to extract facts which can be accurately integrated into a geospatial interface. In evaluation, the model achieves double the precision of a similarly state-of-the-art model trained without the hidden variable while retaining the same level of recall. This improved precision reduces the noise presented to the user in the geospatial interface.

Acknowledgments

We thank Dallan Quass for providing access to a US places and geographic coordinate database. We thank Joe Rogers for help creating our 3D visualization. We thank Microsoft Research for support under the Memex and eScience funding programs and thank Kodak for a gift that helped make this research possible.

This work is supported in part by DoD contract #HM1582-06-1-2013, in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Agichtein, E., and Gravano, L. 2000. Snowball: Extracting relations from large plain-text collections. In *ICDL*.

Baldridge, J.; Morton, T.; and Bierner, G. 2002. The opennlp maximum entropy package. Technical report, SourceForge.

Brin, S. 1998. Extracting patterns and relations from the world-wide web. In *The International Workshop on the Web and Databases*.

Chieu, H. L., and Ng, H. T. 2002. A maximum entropy approach to information extraction from semi-structured and free text. In *AAAI*.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D.; and Yates, A. 2004. Web-scale information extraction in knowitall. In *WWW*.

Flickr. 2006. http://www.flickr.com.

Google Earth. 2006. http://earth.google.com/.

Google Maps. 2006. http://maps.google.com.

Hasegawa, T.; Sekine, S.; and Grishman, R. 2004. Discovering relations amoung named entities from large corpora. In *ACL*.

Kschischang, F.R., F. B., and Loeliger, H. 2001. *IEEE Transactions on Information Theory* 47(2).

Lawrence, N. D., and Scholkopf, B. 2001. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*.

Malouf., R. 2002. A comparison of algorithms for maximum enotrpy parameter estimation.

Mann, G. S., and Yarowsky, D. 2005. Multi-field information extraction and cross-document fusion. In *ACL*.

McCallum, A.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval Journal* 3.

McCallum, A.; Pal, C.; Druck, G.; and Wang, X. 2006. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

Ravichandran, D., and Hovy, E. 2002. Learning surface text patterns for a question answering system. In *ACL*.

Salakhutdinov, R.; Roweis, S. T.; and Ghahramani, Z. 2003. Optimization with em and expectation-conjugate-gradient.

Smith, D. A., and Mann, G. S. 2003. Bootstrapping toponym classifiers. In *The HLT-NAACL Workshop on Analysis of Geographic References*.

Toyama, K.; Logan, R.; Roseway, A.; and Anandan, P. 2003. Geographic location tags on digital images. In *ACM Multimedia*.

Wikimapia. 2006. http://wikimapia.org.