Identifying Patterns in Texts

Minhua Huang
Computer Science, Graduate Center
City University of New York
New York, NY 10016
mhuang@gc.cuny.edu

Robert M. Haralick
Computer Science, Graduate Center
City University of New York
New York, NY 10016
haralick@aim.com

Abstract

We discuss a probabilistic graphical model for recognizing patterns in texts. It is derived from the probability function for a sequence of categories given a sequence of symbols under two reasonable conditional independence assumptions and represented by a product of combinations of conditional and marginal probability functions. The novelty of our model is that it has a mathematical representation which is completely different from existing graphical models such as CRFs, HMMs, and MEMMs. Moreover, it can be used for identifying various patterns in texts. Up to now, we have used this model for recognizing NP chunks and senses of a polysemous word in sentences. This model has achieved very promising results on standard data sets. In the future, we will use this model for extracting semantic roles in a sentence.

1 Introduction

NLP researchers put their efforts on developing methods for extracting patterns in texts. These patterns can be viewed as syntactic patterns or semantic patterns. For example, NP chunks (noun phrases) are syntactic patterns because they are defined by grammatical rules while senses of a polysemous word are semantic patterns because they can be identified by the contexts of the word. Here, we discuss a probabilistic graphical model for recognizing these patterns. The mathematical representation of our model is: $p(c_1, ..., c_N|s_1, ..., s_N) = \prod_{n=1}^{N} p(s_{n-1}|s_n, c_n)\ p(s_{n+1}|s_n, c_n)\ p(s_n|c_n)\ p(c_n)$. It is derived from the probability function of a sequence of categories $(c_1, .., c_N)$ given a sequence of symbols $(s_1, .., s_N)$ where the symbols carry the information in the lexicon and POS tags in a sentence. It has a different perspective compared with the existing graphical models such as CRFs [9], HMMs [16], MEMMs [13].

For identifying these patterns in texts, a sentence is partitioned into several phrases. In the case of the NP chunking problem, these phrases can be NP chunks and OTHERs. In the case of the word sense disambiguation (WSD) problem, each phrase is represented by its last word, called the head word. The context of a polysemous word is represented by a sequence of words. These words may be the last words of phrases or words within $\pm N$ words around the polysemous words. By the model, we automatically assign the category to each word of the sequence having highest probability. We determine NP chunks by grouping consecutive words with the same particular category. We determine the category of the ambiguous word by selecting the most frequent category assigned to that word in the sequence. We may determine a semantic role by first, grouping consecutive words with the same particular category, then categorizing these roles into different classes by designing a set of rules based on the Levin's verb classes.

We test our model for identifying NP chunks with two data sets: the WSJ data set from the Penn Treebank and the CoNLL-2000 shared task data set. Our method achieves an average precision 97.7% and an average recall 98.7% on the first data set and an average precision 95.15% and an average recall 96.05% on the second data set. Moreover, we test our model for WSD by the $line\_serve\_hard\_interest$ data sets. Our model achieves an average of precision 91.38% and an average of recall 91.08% for identifying the $project$ sense of the word $line$; an average of precision 91.36% and an average of recall 90.07% for identifying the $supply\_with\_food$ sense of the word $serve$; an average of precision 86.50% and an average of recall 91.43% for distinguishing the *difficult* sense of the word $hard$; an average of precision 89.50% and an average of recall 91.78% for identifying $the\_money\_paid\_for\_the\_use\_of\_money$ sense of the word $interest$. We are going to test our model for semantic role labeling (SRL) on the CoNLL-2005 shared task data set.

The rest of our discussion is structured in the following way. The second section presents the method. The third section demonstrates the empirical results. The fourth sec-

978-0-7695-3800-6/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSC.2009.22

tion reviews related researches and discussions. The fifth section gives a conclusion.

2 The Proposed Method

2.1 An Example

Table 1 shows the input sentence "*He had deposited his paycheck to the local PNC bank last Saturday morning*." with its POS tags. By the method, each word of the sentence is assigned to one of three different categories C_1, C_2,and C_3. C_1 represents a word inside a block (a block can be a NP chunk or a semantic role), C_2 represents a word outside a block, C_3 represents a word starting a new block. NP chunks or semantic roles are formed by grouping successive words with the same category C_1 or starting with the category C_3 and followed by zero, one, or more consecutive C_1s. The context of the polysemous word *bank* is found by grouping the words corresponding to the last C_1 of consecutive C_1s or last C_2 of consecutive C_2s. Moreover, different semantic roles A_0, A_1, A_2, A_3 are needed to be separated from the semantic roles obtained from the previous step.

Table 1. An example of recognizing text patterns

Lexicon	*POS tag*	*Class NP*	*Class WSD*	*Class SRL*	*NP chunks*	*WSD bank*	*SRL*
He	NNP	C_1	C_1	C_1	1		A_0
had	VBD	C_2	C_2	C_2	1		A_0
deposited	VBN	C_2	C_2	C_2			Verb
his	PRP$	C_1	C_1	C_1	2		A_1
paycheck	NNS	C_1	C_1	C_1	2		A_2
the	DT	C_1	C_1	C_1	3		
local	JJ	C_1	C_1	C_1			
PNC	NN	C_1	C_1	C_1			
bank	NNS	C_1	C_1	C_1	3	$F_{inancial}$	A_2
last	JJ	C_3	C_3	C_3	4		A_3
Saturday	NNP	C_1	C_1	C_1			
morning	NN	C_1	C_1	C_1	4		A_3

In the case of NP chunking, C_1 represents a symbol inside a NP chunk, C_2 represents a symbol outside of a NP chunks,and C_3 represents a symbol starting a new NP chunks. In the case of word sense disambiguation, C_1, C_2, and C_3 are followed the conventions of NP chunking. In the case of semantic role labeling, C_1 represents a symbol inside a semantic role, V represents the main verb, C_2 represents a symbol outside of a semantic role, and C_3 starts a new semantic role.

2.2 Describing the Task

Let L be a language, V be a vocabulary of L, and T be POS tags of V. Let S be a sequence of symbols associated with a sentence, $S = (s_1, ..., s_N)$, where $s_n =< w_n, t_n >$, $w_n \in V$, $t_n \in T$. Let C be a set of categories, $C = \{C_1, C_2, C_3\}$. Let $\mathcal{B}$ be a block. The definition of $\mathcal{B}$ can be found in the section 2.5. C_1 indicates the current symbol is in $\mathcal{B}$, C_2 indicates the current symbol is not in $\mathcal{B}$, and C_3 starts a new $\mathcal{B}$. The tasks can be stated as, given S, we need to find:

1. a sequence of categories, $(c_1, ..., c_N)$, $c_i \in C$, with the best description of S;
2. all the $\mathcal{B}s$ based on $(c_1, ...c_N)$, s.t. $\mathcal{B} = \{B_1, \ldots, B_M\}$, s.t. $B_i \cap B_j = \phi$ and $B_i \subset S$

2.3 Building Graphical Models

Given $S = (s_1, s_2, ..., s_N)$, $C = \{C_1, C_2, C_3\}$, for $s_i \in S$, we want to find $c_i \in C$, s.t.

$$(c_1, c_2, ..., c_N) = \underset{c_1, c_2, ..., c_N}{argmax}\ p(c_1, c_2, ..., c_N|S) \quad (1)$$

Suppose c_i is independent of $c_{j \neq i}$ given $(s_1, s_2, ..., s_N)$. This means that the symbol sequence contains all the information with respect to the category chain associated with any word. Moreover, assume c_i is independent of $(s_1, .., s_{i-2}, s_{i+2}, .., s_N)$ given (s_{i-1}, s_i, s_{i+1}). This means that all the information pertaining to the category class of the word i is in entities contained by the symbol associated with the word i, its predecessor word $i-1$, and its successor word $i+1$. The probability graphical model using these assumptions is shown in Fig 1. From this model, a set of

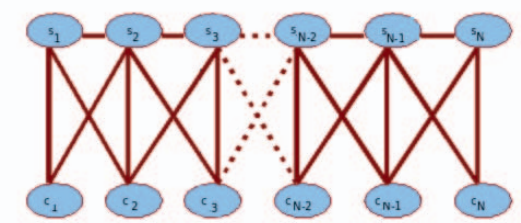

Figure 1. The probabilistic graphical model of $p(c_i|s_1, ...s_N)$ under assumptions c_i is independent of $c_{j \neq i}$ given $(s_1, s_2, ..., s_N)$ and c_i is independent of $(s_1, .., s_{i-2}, s_{i+2}, .., s_N)$ given (s_{i-1}, s_i, s_{i+1})

$2N-2$ cliques[1] is obtained:

$$CIL = \{\{s_1, s_2, c_1\}, \{s_1, s_2, c_2\}, \{s_2, s_3, c_2\}, \ldots, \{s_{N-1}, s_N, c_{N-1}\}, \{s_{N-1}, s_N, c_N\}\}$$

Moreover, there is a corresponding set of $2N-3$ separators [2]:

$$SEP = \{\{s_1, s_2\}, ..., \{s_{N-1}, s_N\}, \{s_2, c_2\}, ..., \{s_{N-1}, c_{N-1}\}\}$$

The junction tree[3] is formed as shown in Fig 2. The cliques are represented as nodes and separators are represented as edges. From this model, according to [2],

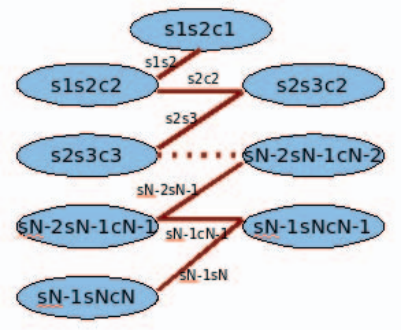

Figure 2. A junction tree for $p(c_1, ..c_N|s_1, .., s_N)$

$p(c_1, .., c_N|s_1, .., s_N)$ can be computed by the product of the probability of the cliques divided by the product of the probabilities of the separators. Hence:

$$p(c_1, ..c_N|s_1, .., s_N) = M_{\mathcal{S}} \prod_{n=1}^{N} p(s_{n-1}|s_n, c_n)p(s_{n+1}|s_n, c_n)p(s_n|c_n)p(c_n) \quad (2)$$

In (2), we define $p(s_0|s_1, c_1) = p(s_{N+1}|s_N, c_N) = 1$ and $M_{s_1,...,s_M}$ is a constant depending only on $s_1, .., s_n$ and not depending on any c_n. It can be obtained by the equation (3).

$$M_{\mathcal{S}} = \frac{1}{p(s_1, s_2)p(s_2, s_3)...p(s_{N-1}, s_N)} \quad (3)$$

2.4 Making Decisions

Because each c_n is independent of each other in (2), its value can be determined individually. We can find c_n by $c_n = argmax_{c_n}\ p(s_{n-1}|s_n, c_n)\ p(s_{n+1}|s_n, c_n)\ p(s_n|c_n)\ p(c_n)$. Then, (1) can be rewritten as:

$$\begin{aligned}
& (c_1, c_2, ..., c_N) \\
= \ & (\underset{c_1}{argmax}(p(s_2|s_1, c_1)p(s_1|c_1)p(c_1)), \\
& \underset{c_2}{argmax}(p(s_1|s_2, c_2)p(s_3|s_2, c_2)p(s_2|c_2)p(c_2)), \\
& ..., \\
& \underset{c_{N-1}}{argmax}(p(s_{N-2}|s_{N-1}, c_{N-1})p(s_N|s_{N-1}, c_{N-1}) \\
& p(s_{N-1}|c_{N-1})p(c_{N-1})), \\
& \underset{c_N}{argmax}(p(s_{N-1}|s_N, c_N)p(s_N|c_N)p(c_N))) \quad (4)
\end{aligned}$$

2.5 Determining NP Chunks or Semantic Roles

Given $(< s_1, c_1 >, \ldots, < s_N, c_N >)$, $s_i \in S$, $c_i \in C$, S, C are defined in the section 2.2 . Let $\mathcal{B}$ be a block if and only if:

1. for some $i < j$, $\mathcal{B} = (< s_i, c_i >, < s_{i+1}, c_{i+1} > , \ldots, < s_j, c_j >)$
2. $c_i \in \{C_1, C_3\}$
3. $c_n = C_1, \quad n = i+1, \ldots, j$
4. $\mathcal{B}' \subseteq \mathcal{B}$ and $\mathcal{B}'$ satisfies (1), (2), and (3) $\Rightarrow \mathcal{B}' = \mathcal{B}$

2.6 Determining the Sense of a Polysemous Word

Given $(< s_1, c_1 >, \ldots, < s_N, c_N >)$, $s_i \in S$, $c_i \in C$, S, C are defined in the section 2.2 . We assign the class C_k for the polysemous word w_t if and only if:

$$\#\{n \in \{1, \ldots, N\}|c_n = C_k\} > \#\{m \in \{1, \ldots, N\}|c_m = C_j\}, \quad i \neq k \quad (5)$$

2.7 Estimating Probabilities of the Model

We use a training set to estimate the probabilities for the equation (2). In our model, the probability of a current symbol being assigned to the class c in a sequence associated with a sentence is partially dependent on the probability of the previous symbol given the current symbol and the class c, and the probability of the success symbol given the current symbol and the class c. In this way, the adjacency between two neighboring symbols of an incoming sentence are preserved by the overlapping of our model. Therefore, we can consider a group of k sentences for estimating the

[1] a clique is a maximal complete set of nodes.

[2] $\Gamma = \{\Gamma_1, \ldots, \Gamma_M\}$ is a set of separators, where $\Gamma_k = \Lambda_k \cap (\Lambda_1 \cup ..., \cup \Lambda_{k-1})$

[3] A junction tree is a maximum spanning tree w.r.t separator size. A maximum spanning tree is a tree over V whose sum of edge weights has a maximum value. Here the edge weights are the sizes of the separators.

probabilities. Our training set has K sentences. Each sentence k consists of N_k words, $< x_{k,1}, .., x_{k,N_k} >$, and the corresponding class labels $< y_{k,1}, .., y_{k,N_k} >$. Hence the training set is $\Psi = \{\psi_1, \psi_2, ..., \psi_K\}$, where $\psi_k = (< x_{k,1}, y_{k,1} >, ..., < x_{k,N_k}, y_{k,N_k} >)$. Let t, w, z, c be random variables, where w designates a word, t designates a word before w, z designates a word after w, and c designates a class. Let $p_{w|c}(\alpha|\gamma)$ designate the conditional probability of a word being α given that its class is γ. Let $p_{t|w,c}(\alpha|\beta,\gamma)$ designate the conditional probability of the word previous to the current word being α given that the current word is β and its class is γ. Let $p_{z|w,c}(\alpha|\beta,\gamma)$ designate the conditional probability of the word after the current word being α given that the current word is β and its class is γ.
Let:

$$I = \{(k,n)|k = 1, \ldots, K, n = 1, \ldots, N_k\}$$

$p_{w|c}(\alpha|\gamma)$ can be estimated by:

$$\hat{p}_{w|c}(\alpha|\gamma) = \frac{\#\{(k,n) \in I|\alpha = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k,n) \in I|\gamma = y_{k,n}\}} \quad (6)$$

$p_{t|w,c}(\alpha|\beta,\gamma)$ can be estimated by:

$$\hat{p}_{t|w,c}(\alpha|\beta,\gamma) = \frac{\#\{(k,n) \in I|\alpha = x_{k,n-1}, \beta = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k,n) \in I|\beta = x_{k,n}, \gamma = y_{k,n}\}} \quad (7)$$

$p_{z|w,c}(\alpha|\beta,\gamma)$ can be estimated by:

$$\hat{p}_{z|w,c}(\alpha|\beta,\gamma) = \frac{\#\{(k,n) \in I|\alpha = x_{k,n+1}, \beta = x_{k,n}, \gamma = y_{k,n}\}}{\#\{(k,n) \in I|\beta = x_{k,n}, \gamma = y_{k,n}\}} \quad (8)$$

3 Empirical Results

We define features based on the equation (2)as follows:

$$f(s_i, c_i) = p(c_i)q(s_i|c_i)r(s_{i-1}|s_i, c_i)t(s_{i+1}|s_i, c_i) \quad (9)$$

We form the training set by including 90% instances and the testing set by including 10% instances of the whole data set. In this way, we do it iteratively 10 times to select the different training sets and testing sets. The evaluation metric we have used are precision p_{re}, recall R_{ec}, and f-measure $f_{me} = \frac{2*p_{re}*R_{ec}}{P_{re}+R_{ec}}$.

3.1 Identifying NP Chunks Using CoNLL-2000 Shared Task Data Set

We have conducted three different tests on the CoNLL-2000 shared task data set by choosing different values of features. From these selections, we examine the probabilities in order to find the one which contributes the best performance. In the first test, we include all lexicon and POS tags from the data set. We have tested our model according to descriptions in the section 3. We averaged the results that we received. The average precision is 95.15%, the average recall is 96.05%, and the average f-measure is 95.59%. In the second test, the lexicon is excluded. We only include the POS tags. In the third test, all POS tags are excluded and only the lexicon is included. The results are shown in the table 2. By comparing the three results on the CoNLL-2000 shared task data, we have noticed that if the model is built only on the lexical information, it has the lowest performance of f-measure 89.75%. The model's performance improved 3% in f-measure if it is constructed by POS tags. The model achieves the best performance of 95.59% in f-measure if we are considering both lexicons and POS tags.

Table 2. The results on the CoNLL-2000 data

Measurement	Lexicon + POS tags %	POS tags %	Lexicon %
P_{re}	95.15	92.27	86.42
R_{ec}	96.05	93.76	93.35
F_{me}	95.59	92.76	89.75

The best performances have achieved on the feature values containing lexicon + POS tags. In this case, the average f-measure is 95.59%.

3.2 Identifying NP Chunks Using WSJ Data Set from Penn Treebank

The second data set on which we have experimented is the WSJ data of Penn Treebank. The main reason for us to use this data set is that we want to see whether the performance of our model can be improved when it is built on more data. We build our model on a training set which is seven times larger than the CoNLL-2000 shared task training data set (Section 3.3). The performance of our method for the data is listed in the table 3. The average precision is increased 2.7% from 95.15% to 97.73%. The average recall is increased 2.8% from 96.05% to 98.65%. The average f-measure is increased 2.7% from 95.59% to 98.2%.

3.3 Identifying Sense of Polysemous Words Using $line\_interest\_hard\_serve$ Data Sets

We test our model for WSD on the data sets $line$, $hard$, $serve$, and $interest$. The senses' descriptions and in-

Table 3. The test results on the WSJ data from the Penn Treebank

Training 800 files	Testing 100 files	P_{re}	R_{ec}	F_{me}
200-999	1100-1199	0.9806	0.9838	0.9822
	1200-1299	0.9759	0.9868	0.9814
	1300-1399	0.9794	0.9863	0.9828
	1400-1499	0.9771	0.9868	0.9817
	1500-1599	0.9768	0.9858	0.9814
	1600-1699	0.9782	0.9877	0.9829
	1700-1799	0.9770	0.9877	0.9824
	1800-1899	0.9771	0.9848	0.9809
	1900-1999	0.9774	0.9863	0.9819
	2000-2099	0.9735	0.9886	0.9806
μ		0.9773	0.9865	0.9818
σ		0.0019	0.0014	0.0008

The recall, precision, and f-measure obtained for each test of 100 files. The average recall, precision, and f-measure and their standard deviations are obtained from 1000 testing files.

Table 4. The results on *line, hard, serve, interest* **data**

Word	Sense Description	# of Instance	f_{me} %
$line$	*project*	2218	92.24
$noun$	*phone*	429	85.22
	text	404	81.95
$hard$	*difficult*	3345	88.76
adj	*not soft*	502	83.75
	physical not soft	376	80.05
$serve$	*supply with food*	1841	91.04
$verb$	*hold an office*	1272	87.48
	function as something	853	82.52
$interst$	*money paid for the use of money*	1272	91.45
$noun$	*a share in a company*	500	88.85
	readiness to give attention	361	79.95

stances' distributions can be found in [11] and [3]. Because of the limitations of number of instances (a sentence having the polysemous word in it) for each sense in the corpora, we select the first three senses for each polysemous word in our test. Again, the values of features are made by lexicon + POS tags. We test our model based the descriptions of the section 3. The test results are shown in the table 4. We have noticed that, with the same instances for a polysemous noun, adjective, or verb, our model achieves the best f measure for polysemous nouns and the worst result for polysemous adjectives. For example, the average $f_{me} > 91\%$ if the number of instances > 1270 for the polysemous nouns $line$ and $interest$. However, in order to keep the same f-measure value, the polysemous word $serve$ needs to have more instance: 1841 instances. The polysemous adjective needs to have about 3350 instances to reach the average $f_{me} = 88.76\%$. Moreover, in any case, our model achieves an average $f_{ma} = 80\%$ if the number of instances is reduced to about 400. We conclude that the performance of our model on WSD is dependent on the number of instances in the training set: the larger the better.

4 Related Research and Discussion

Currently existing graphical models for NLP are HMMs[13] [16], MEMMs[13], and CRFs[9] [18]. These models are built under different conditional independence assumptions for obtaining the sequence $< c_1, \ldots, c_N >$ that maximizes $p(c_1, \ldots, c_N, s_1, \ldots, s_N)$ or $p(c_1, \ldots, c_N | s_1, \ldots, s_N)$. Among these models, HMMs and MEMMs are directed graphic models while CRFs and our model are undirected graphical models. Comparing with these two undirected graphic models, each c_i links to c_{i-1} and s_i in CRFs. Therefore, c_i is dependent on c_{i-1} and s_i. In contract, in our model, each c_i links to s_i, s_{i-1}, and s_{i+1}, not the previous category c_{i-1}. Therefore, c_i is not dependent on c_{i-1}. This makes it possible for the sequence $< c_1, .., c_N >$ with the maximum value of $p(c_1, \ldots, c_N | s_1, \ldots, s_N)$ be determined by finding each c_i that satisfies the equation (4). In this way, the time complexity for recognizing a new incoming sequence with N symbols at the worst case is $M * N$, where M is the number of categories. For example, if $C = \{C_1, C_2, C_3\}$, then $M = 3$. Therefore, the time complexity is $O(N)$. The memory space also will be reduced compared the other graphic models because we donet need to store all the previous category chains into the memory. Moreover, our model is more reliable and stable due to the global maximum probability being obtained by the local maximal probabilities. There is no a chance to change the previous category chain because of an accidently higher probability at the current state.

A number of NP chunking and WSD methods have been developed over the years. The methods for NP chunking are [4] [17] [16] [18] [20] while the methods for WSD are [6] [5] [11] [10] [21]. Our method adopt Ramshaw's idea [17] of assigning different categories to words in a sentence based on whether these words are inside a NP chunk, outside a NP chunk, or start a new NP chunk. For WSD , in

contrast with other methods, the polysemous word is represented by a sequences of ordered words with POS tags. Our model assigns a category for a word of the input sentence based on the information of the word, the previous word, and the next word we have met before, which a human often does this in the same way. The experiments in the section 3.1 show our model achieves better performance than HMMs and CRFs [18]. For WSD, our method can achieve precision and recall > 90 if number of instances of a sense ≥ 1000.

5 Conclusions

Recognizing patterns in a sentence is the first step toward understanding the meaning of the sentence. This paper presents a new probabilistic graphical model for doing such tasks. Experiments show that our model is effective. We have achieved an average of precision 97.7% and an average of recall 98.7% on WSJ data from the Penn Treebank and an average precision 95.15% and an average recall 96.05% on CoNLL-2000 shared task data set for recognizing NP chunks in a sentence. Moreover, we have achieved an average precision 90.57% and an average of recall 92.35% on recognizing a particular sense of polysemous nouns, an average precision 90.86% and an average recall 91.22% on recognizing a particular sense of a polysemous verb, and an average precision 86.50% and an average recall 91.01% on recognizing a particular sense of a polysemous adjective. From the empirical results, in order to improve the performance of WSD, we need to increase the size of the training set. In the future, we will expend number of instances for $line\_interest\_hard\_serve$ data sets and test our model for other senses of these polysemous words. Moreover, we will test our model on recognizing the semantic roles in a sentence by using CoNLL-2005 shared task data set.

References

[1] S. Abney and S. P. Abney. Parsing by chunks. In *Principle-Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2002.

[3] R. Bruce and J. Wiebe. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 139–146, 1994.

[4] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136 – 143, 1988.

[5] W. Gale, K. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. In *Computers and the Humanities*, pages 415–439, 1992.

[6] M. A. Hearst. Noun homograph disambiguation using local context in large text corpora. In *Proceedings of the Seventh Annual Conference of the UW centre for the New OED and Text Research*, pages 1–22, 1991.

[7] M. Huang and R. M. Haralick. A graphical model for recognizing noun phreases from text. In *Preceedings of the* $3^r d$ *internationa conference on Language and Automata Theory and Applications*, 2009.

[8] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. AI Pearson Education, 2006.

[9] J. Lafferty, A. MaCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[10] C. Leacock, G. A. Miller, and M. Chodorow. Using corpus statistics and wordnet relations for sense identification. *Computational Linguist.*, 24:147–165, 1998.

[11] C. Leacock, G. Towell, and E. Voorhees. Corpus based statistical sense resolution. In *Proceedings of the workshop on Human Language Technology*, pages 260 – 265, 1993.

[12] E. Levin, M. Sharifi, and J. Ball. Evaluation of utility of lsa for word sense discrimination. In *Preceedings of HLT-NAACL*, pages 77 – 80, 2006.

[13] A. MaCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of 17th International Conf. on Machine Learning*, pages 591–598, 2000.

[14] C. Manning and H. Schutze. *Foundations of statistical natural language processing*. The MIT Press Cambridge, 2003.

[15] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[16] A. Molina, F. Pla, D. D. S. Informtics, J. Hammerton, M. Osborne, S. Armstrong, and W. Daelemans. Shallow parsing using specialized hmms. *Journal of Machine Learning Research*, 2:595–613, 2002.

[17] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, 1995.

[18] F. Sha and F. Fereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, pages 213–220, 2003.

[19] E. F. Tjong and K. Sang. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000*, pages 127–132, 2000.

[20] Wu-Chieh, Wu, Y.-S. Lee, and J.-C. Yang. Robust and efficient multiclass svm models for phrase pattern recognition. *Pattern Recognition*, 41:2874–2889, 2008.

[21] D. Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and frech. In *Preceedings of the 32nd Annual Meeting*, 1994.

[22] D. Yarowsky and R. Florian. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and frech. *Natural Language Engineering*, pages 293–310, 2002.