Structured Generative Models for Unsupervised Named-Entity Clustering

Micha Elsner, Eugene Charniak and Mark Johnson
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI 02912
{melsner,ec,mj}@cs.brown.edu

Abstract

We describe a generative model for clustering named entities which also models named entity internal structure, clustering related words by role. The model is entirely unsupervised; it uses features from the named entity itself and its syntactic context, and coreference information from an unsupervised pronoun resolver. The model scores 86% on the MUC-7 named-entity dataset. To our knowledge, this is the best reported score for a fully unsupervised model, and the best score for a generative model.

1 Introduction

Named entity clustering is a classic task in NLP, and one for which both supervised and semi-supervised systems have excellent performance (Mikheev et al., 1998; Chinchor, 1998). In this paper, we describe a fully unsupervised system (using no "seed rules" or initial heuristics); to our knowledge this is the best such system reported on the MUC-7 dataset. In addition, the model clusters the words which appear in named entities, discovering groups of words with similar roles such as first names and types of organization. Finally, the model defines a notion of consistency between different references to the same entity; this component of the model yields a significant increase in performance.

The main motivation for our system is the recent success of unsupervised generative models for coreference resolution. The model of Haghighi and Klein (2007) incorporated a latent variable for named entity class. They report a named entity score of 61.2 percent, well above the baseline of 46.4, but still far behind existing named-entity systems.

We suspect that better models for named entities could aid in the coreference task. The easiest way to incorporate a better model is simply to run a supervised or semi-supervised system as a preprocess. To perform joint inference, however, requires an unsupervised generative model for named entities. As far as we know, this work is the best such model.

Named entities also pose another problem with the Haghighi and Klein (2007) coreference model; since it models only the heads of NPs, it will fail to resolve some references to named entities: ("Ford Motor Co.", "Ford"), while erroneously merging others: ("Ford Motor Co.", "Lockheed Martin Co."). Ng (2008) showed that better features for matching named entities– exact string match and an "alias detector" looking for acronyms, abbreviations and name variants– improve the model's performance substantially. Yet building an alias detector is nontrivial (Uryupina, 2004). English speakers know that "President Clinton" is the same person as "Bill Clinton" , not "President Bush". But this cannot be implemented by simple substring matching. It requires some concept of the role of each word in the string. Our model attempts to learn this role information by clustering the words within named entities.

2 Related Work

Supervised named entity recognition now performs almost as well as human annotation in English (Chinchor, 1998) and has excellent performance on other languages (Tjong Kim Sang and De Meulder, 2003). For a survey of the state of the art,

see Nadeau and Sekine (2007). Of the features we explore here, all but the pronoun information were introduced in supervised work. Supervised approaches such as Black et al. (1998) have used clustering to group together different nominals referring to the same entity in ways similar to the "consistency" approach outlined below in section 3.2.

Semi-supervised approaches have also achieved notable success on the task. Co-training (Riloff and Jones, 1999; Collins and Singer, 1999) begins with a small set of labeling heuristics and gradually adds examples to the training data. Various co-training approaches presented in Collins and Singer (1999) all score about 91% on a dataset of named entities; the inital labels were assigned using 7 hand-written seed rules. However, Collins and Singer (1999) show that a mixture-of-naive-Bayes generative clustering model (which they call an EM model), initialized with the same seed rules, performs much more poorly at 83%.

Much later work (Evans, 2003; Etzioni et al., 2005; Cucerzan, 2007; Pasca, 2004) relies on the use of extremely large corpora which allow very precise, but sparse features. For instance Etzioni et al. (2005) and Pasca (2004) use web queries to count occurrences of "cities such as X" and similar phrases. Although our research makes use of a fairly large amount of data, our method is designed to make better use of relatively common contextual features, rather than searching for high-quality semantic features elsewhere.

Models of the internal structure of names have been used for cross-document coreference (Li et al., 2004; Bhattacharya and Getoor, 2006) and a goal in their own right (Charniak, 2001). Li et al. (2004) take named entity classes as a given, and develops both generative and discriminative models to detect coreference between members of each class. Their generative model designates a particular mention of a name as a "representative" and generates all other mentions from it according to an editing process. Bhattacharya and Getoor (2006) operates only on authors of scientific papers. Their model accounts for a wider variety of name variants than ours, including misspellings and initials. In addition, they confirm our intuition that Gibbs sampling for inference has insufficient mobility; rather than using a heuristic algorithm as we do (see section 3.5), they use a data-driven block sampler. Charniak (2001) uses a Markov chain to generate 6 different components of people's names, again assuming that the class of personal names can be pre-distinguished using a name list. He infers coreference relationships between similar names appearing in the same document, using the same notion of consistency between names as our model. As with our model, the clusters found are relatively good, although with some mistakes even on frequent items (for example, "John" is sometimes treated as a descriptor like "Secretary").

3 System Description

Like Collins and Singer (1999), we assume that the named entities have already been correctly extracted from the text, and our task is merely to label them. We assume that all entities fit into one of the three MUC-7 categories, LOC (locations), ORG (organizations), and PER (people). This is an oversimplification; Collins and Singer (1999) show that about 12% of examples do not fit into these categories. However, while using the MUC-7 data, we have no way to evaluate on such examples.

As a framework for our models, we adopt *adaptor grammars* (Johnson et al., 2007), a framework for non-parametric Bayesian inference over context-free grammars. Although our system does not require the full expressive power of PCFGs, the adaptor grammar framework allows for easy development of structured priors, and supplies a flexible generic inference algorithm. An adaptor grammar is a hierarchical Pitman-Yor process (Pitman and Yor, 1997). The grammar has two parts: a base PCFG and a set of *adapted* nonterminals. Each adapted nonterminal is a Pitman-Yor process which expands either to a previously used subtree or to a sample from the base PCFG. The end result is a posterior distribution over PCFGs and over parse trees for each example in our dataset.

Each of our models is an adaptor grammar based on a particular base PCFG where the top nonterminal of each parse tree represents a named entity class.

3.1 Core NP Model

We begin our analysis by reducing each named-entity reference to the contiguous substring of

$$
\begin{aligned}
ROOT &\rightarrow NE_0 | NE_1 | NE_2 \\
NE_0 &\rightarrow (NE_0^0)(NE_0^1)(NE_0^2)(NE_0^3)(NE_0^4) \\
*NE_0^0 &\rightarrow Words \\
*Words &\rightarrow Word\ (Words) \\
Word &\rightarrow Bill \ldots
\end{aligned}
$$

Figure 1: Part of the grammar for core phrases. (Parentheses) mark optional nonterminals. *Starred nonterminals are adapted.

$$
\begin{aligned}
ROOT &\rightarrow NE_0 | NE_1 | NE_2 \\
NE_0 &\rightarrow E_{00} | E_{01} \ldots E_{0k} \\
E_{00} &\rightarrow (E_{00}^0)(E_{00}^1)(E_{00}^2)(E_{00}^3)(E_{00}^4) \\
**E_{00}^0 &\rightarrow NE_0^0 \\
*NE_0^0 &\rightarrow Words \ldots
\end{aligned}
$$

Figure 2: Part of the consistency-enforcing grammar for core phrases. There are an infinite number of entities E_{xk}, all with their own lexical symbols. Each lexical symbol E_{xk}^i expands to a single NE_x^i.

proper nouns which surrounds its head, which we call the core (Figure 1). To analyze the core, we use a grammar with three main symbols (NE_x), one for each named entity class x. Each class has an associated set of lexical symbols, which occur in a strict order (NE_x^i is the ith symbol for class x). We can think of the NE^i as the semantic parts of a proper name; for people, NE_{PER}^0 might generate titles and NE_{PER}^1 first names. Each NE^i is adapted, and can expand to any string of words; the ability to generate multiple words from a single symbol is useful both because it can learn to group collocations like "New York" and because it allows the system to handle entities longer than four words. However, we set the prior on multi-word expansions very low, to avoid degenerate solutions where most phrases are analyzed with a single symbol. The system learns a separate probability for each ordered subset of the NE^i (for instance the rule $NE_0 \rightarrow NE_0^0\, NE_0^2\, NE_0^4$), so that it can represent constraints on possible references; for instance, a last name can occur on its own, but not a title.

3.2 Consistency Model

This system captures some of our intuitions about core phrases, but not all: our representation for "Bill Clinton" does not share any information with "President Bill Clinton" except the named-entity class. To remedy this, we introduce a set of "entity" nonterminals E_k, which enforce a weak notion of consistency. We follow Charniak (2001) in assuming that two names are consistent (can be references to the same entity) if they do not have different expansions for any lexical symbol. In other words, a particular entity $E_{PER,Clinton}$ has a title $E_{PER,Clinton}^0 =$ "President", a first name $E_{PER,Clinton}^1 =$ "Bill" etc. These are generated from the class-specific distributions, for instance $E_{PER,Clinton}^0 \sim E_{PER}^0$, which we intend to be a distribution over titles in general.

The resulting grammar is shown in Figure 2; the prior parameters for the entity-specific symbols E_{xk}^i are fixed so that, with overwhelming probability, only one expansion occurs. We can represent any fixed number of entities E_k with a standard adaptor grammar, but since we do not know the correct number, we must extend the adaptor model slightly to allow for an unbounded number. We generate the E_k from a Chinese Restaurant process prior. (General grammars with infinite numbers of nonterminals were studied by (Liang et al., 2007b)).

3.3 Modifiers, Prepositions and Pronouns

Next, we introduce two types of context information derived from Collins and Singer (1999): nominal modifiers and prepositional information. A nominal modifier is either the head of an appositive phrase ("Maury Cooper, a vice **president**") or a non-proper prenominal ("**spokesman** John Smith")[1]. If the entity is the complement of a preposition, we extract the preposition and the head of the governing NP ("a federally funded sewage **plant in** Georgia"). These are added to the grammar at the named-entity class level (separated from the core by a special punctuation symbol).

Finally, we add information about pronouns and wh-complementizers (Figure 3). Our pronoun information is derived from an unsupervised coreference algorithm which does not use named entity informa-

[1] We stem modifiers with the Porter stemmer.

$$
\begin{aligned}
ROOT \rightarrow & Modifiers_0 \ \#\ NE_0\ \# \\
& \quad Prepositions_0\ \#\ Pronouns_0\ \# \\
& \ldots \\
Pronouns_0 \rightarrow & Pronoun_0\ Pronouns_0 \\
Pronouns_0 \rightarrow & \\
Pronoun_0 \rightarrow & \mathbf{pers} | loc | org | any \\
pers \rightarrow & i | he | she | who | me \ldots \\
loc \rightarrow & where | which | it | its \\
org \rightarrow & which | it | they | we \ldots
\end{aligned}
$$

Figure 3: A fragment of the full grammar. The symbol # represents punctuation between different feature types. The prior for class 0 is concentrated around **personal** pronouns, although other types are possible.

attack airlift airlift rescu # wing # of-commander of-command with-run # #
air-india # #
abels # # it
gaudreau # # they he
priddy # # he
spokesman bird bird bird director bird ford clinton director bird # johnson # before-hearing to-happened of-cartoon on-pressure under-medicare to-according to-allied with-stuck of-government of-photographs of-daughter of-photo for-embarrassing under-instituted about-allegations for-worked before-hearing to-secretary than-proposition of-typical # he he his he my himself his he he he he i he his his i i i he his #

Figure 4: Some merged examples from an input file. (# separates different feature types.)

tion (Charniak and Elsner, 2009). This algorithm uses EM to learn a generative model with syntactic, number and gender parameters. Like Haghighi and Klein (2007), we give our model information about the basic types of pronouns in English. By setting up the base grammar so that each named-entity class prefers to associate to a single type of pronoun, we can also determine the correspondence between our named-entity symbols and the actual named-entity labels– for the models without pronoun information, this matching is arbitrary and must be inferred during the evaluation process.

3.4 Data Preparation

To prepare data for clustering with our system, we first parse it with the parser of Charniak and Johnson (2005). We then annotate pronouns with Charniak and Elsner (2009). For the evaluation set, we use the named entity data from MUC-7. Here, we extract all strings in <ne> tags and determine their cores, plus any relevant modifiers, governing prepositions and pronouns, by examining the parse trees. In addition, we supply the system with additional data from the North American News Corpus (NANC). Here we extract all NPs headed by proper nouns.

We then process our data by merging all examples with the same core; some merged examples from our dataset are shown in Figure 4. When two examples are merged, we concatenate their lists of modifiers, prepositions and pronouns (capping the length of each list at 20 to keep inference tractable). For instance, "air-india" has no features outside the core, while "wing" has some nominals ("attack" &c.) and some prepositions ("commander-of" &c.). This merging is useful because it allows us to do inference based on types rather than tokens (Goldwater et al., 2006). It is well known that, to interpolate between types and tokens, Hierarchical Dirichlet Processes (including adaptor grammars) require a deeper hierarchy, which slows down inference and reduces the mobility of sampling schemes. By merging examples, we avoid using this more complicated model. Each merged example also represents many examples from the training data, so we can summarize features (such as modifiers) observed throughout a large input corpus while keeping the size of our input file small.

To create an input file, we first add all the MUC-7 examples. We then draw additional examples from NANC, ranking them by how many features they have, until we reach a specified number (larger datasets take longer, but without enough data, results tend to be poor).

3.5 Inference

Our implementation of adaptor grammars is a modified version of the Pitman-Yor adaptor grammar

sampler[2], altered to deal with the infinite number of entities. It carries out inference using a Metropolis-within-Gibbs algorithm (Johnson et al., 2007), in which it repeatedly parses each input line using the CYK algorithm, samples a parse, and proposes this as the new tree.

To do Gibbs sampling for our consistency-enforcing model, we would need to sample a parse for an example from the posterior over every possible entity. However, since there are thousands of entities (the number grows roughly linearly with the number of merged examples in the data file), this is not tractable. Instead, we perform a restricted Gibbs sampling search, where we enumerate the posterior only for entities which share a word in their core with the example in question. In fact, if the shared word is very common (occuring in more than .001 of examples), we compute the posterior for that entity only .05 of the time[3]. These restrictions mean that we do not compute the exact posterior. In particular, the actual model allows entities to contain examples with no words in common, but our search procedure does not explore these solutions.

For our model, inference with the Gibbs algorithm seems to lack mobility, sometimes falling into very poor local minima from which it does not seem to escape. This is because, if there are several references to the same named entity with slightly different core phrases, once they are all assigned to the wrong class, it requires a low-probability series of individual Gibbs moves to pull them out. Similarly, the consistency-enforcing model generally does not fully cluster references to common entities; there are usually several "Bill Clinton" clusters which it would be best to combine, but the sequence of moves that does so is too improbable. The data-merging process described above is one attempt to improve mobility by reducing the number of duplicate examples. In addition, we found that it was a better use of CPU time to run multiple samplers with different initialization than to perform many iterations. In the experiments below, we use 20 chains, initializing with 50 iterations without using consistency, then 50 more using the consistency model, and evaluate the last sample from each. We discard the 10 samples with worst log-likelihood and report the average score for the other 10.

3.6 Parameters

In addition to the base PCFG itself, the system requires a few hyperparameter settings: Dirichlet priors for the rule weights of rules in the base PCFG. Pitman-Yor parameters for the adapted nonterminals are sampled from vague priors using a slice sampler (Neal, 2003). The prior over core words was set to the uniform distribution (Dirichlet 1.0) and the prior for all modifiers, prepositions and pronouns to a sparse value of .01. Beyond setting these parameters to a priori reasonable values, we did not optimize them. To encourage the system to learn that some lexical symbols were more common than others, we set a sparse prior over expansions to symbols[4]. There are two really important hyperparameters: an extremely biased prior on class-to-pronoun-type probabilities (1000 for the desired class, .0001 for everything else), and a prior of .0001 for the $Word \rightarrow Word\ Words$ rule to discourage symbols expanding to multiword strings.

4 Experiments

We performed experiments on the named entity dataset from MUC-7 (Chinchor, 1998), using the training set as development data and the formal test set as test data. The development set has 4936 named entities, of which 1575 (31.9%) are locations, 2096 (42.5%) are organizations and 1265 (25.6%) people. The test set has 4069 named entities, 1321 (32.5%) locations, 1862 (45.8%) organizations and 876 (21.5%) people[5]. We use a baseline which gives all named entities the same label; this label is mapped to "organization".

In most of our experiments, we use an input file of 40000 lines. For dev experiments, the labeled data contributes 1585 merged examples; for test experiments, only 1320. The remaining lines are derived

[2]Available at http://www.cog.brown.edu/ mj/Software.htm

[3]We ignore the corresponding Hastings correction, as in practice it leads to too many rejections.

[4]Expansions that used only the middle three symbols $NE_x^{1,2,3}$ got a prior of .005, expansions whose outermost symbol was $NE_x^{0,4}$ got .0025, and so forth. This is not so important for our final system, which has only 5 symbols, but was designed during development to handle systems with up to 10 symbols.

[5]10 entities are labeled location|organization; since this fraction of the dataset is insignificant we score them as wrong.

Model	Accuracy
Baseline (All Org)	42.5
Core NPs (no consistency)	45.5
Core NPs (consistency)	48.5
Context Features	83.3
Pronouns	87.1

Table 1: Accuracy of various models on development data.

Model	Accuracy
Baseline (All Org)	45.8
Pronouns	86.0

Table 2: Accuracy of the final model on test data.

using the process described in section 3.4 from 5 million words of NANC.

To evaluate our results, we map our three induced labels to their corresponding gold label, then count the overlap; as stated, this mapping is predictably encoded in the prior when we use the pronoun features. Our experimental results are shown in Table 1. All models perform above baseline, and all features contribute significantly to the final result. Test results for our final model are shown in Table 2.

A confusion matrix for our highest-likelihood test solution is shown as Figure 5. The highest confusion class is "organization", which is confused most often with "location" but also with "person". "location" is likewise confused with "organization". "person" is the easiest class to identify– we believe this explains the slight decline in performance from dev to test, since dev has proportionally more people.

Our mapping from grammar symbols to words appears in Table 3; the learned prepositional and modifier information is in Table 4. Overall the results are good, but not perfect; for instance, the $Pers$ states are mostly interpretable as a sequence of title - first name - middle name or initial - last name - last name or post-title (similar to (Charniak, 2001)). The organization symbols tend to put nationalities and other modifiers first, and end with institutional types like "inc." or "center", although there is a similar (but smaller) cluster of types at Org^2, suggesting the system has incorrectly found two analyses for these names. Location symbols seem to put entities with a single, non-analyzable name into Loc^2, and use symbols 0, 1 and 3 for compound names. Loc^4 has been recruited for time expressions, since our NANC dataset includes many of these, but we failed to account for them in the model. Since they appear in a single class here, we are optimistic that they could be clustered separately if another class and some appropriate features were added to the prior. Some errors do appear ("supreme court" and "house" as locations, "minister" and "chairman" as middle names, "newt gingrich" as a multiword phrase). The table also reveals an unforeseen issue with the parser: it tends to analyze the dateline beginning a news story along with the following NP ("WASHINGTON Bill Clinton said..."). Thus common datelines ("washington", "new york" and "los angeles") appear in state 0 for each class.

	loc	*org*	*per*
LOC	1187	97	37
ORG	223	1517	122
PER	36	20	820

Figure 5: Confusion matrix for highest-likelihood test run. Gold labels in CAPS, induced labels *italicized*. Organizations are most frequently confused.

5 Discussion

As stated above, we aim to build an unsupervised generative model for named entity clustering, since such a model could be integrated with unsupervised coreference models like Haghighi and Klein (2007) for joint inference. To our knowledge, the closest existing system to such a model is the EM mixture model used as a baseline in Collins and Singer (1999). Our system improves on this EM system in several ways. While they initialize with minimal supervision in the form of 7 seed heuristics, ours is fully unsupervised. Their results cover only examples which have a prepositional or modifier feature; we adopt these features from their work, but label all entities in the predefined test set, including those that appear without these features. Finally, as discussed, we find the "person" category to be the easiest to label. 33% of the test items in Collins and Singer (1999) were people, as opposed to 21% of ours. However, even without the pronoun features, that is, using the same feature set, our system scores equivalently to the EM model, at 83% (this score is

$Pers^0$	$Pers^1$	$Pers^2$	$Pers^3$	$Pers^4$
rep.	john (767)	minister	brown	jr.
sen. (256)	robert (495)	j.	smith (97)	a
washington	david	john (242)	b	smith (111)
dr.	michael	l.	johnson	iii
los angeles	james	chairman	newt gingrich	williams
senate	president	e.	king	wilson
house	richard	m.	miller	brown
new york	william (317)	william (173)	kennedy	clinton
president	sen. (236)	robert (155)	martin	simpson
republican	george	r.	davis	b
Org^0	Org^1	Org^2	Org^3	Org^4
american (137)	national	university	research	association
washington	american (182)	inc. (166)	medical	center
washington the	new york	corp. (156)	news	inc. (257)
national	international (136)	college	health	corp. (252)
first	public	institute (87)	services	co.
los angeles	united	group	communications	committee
new	house	hospital	development	institute
royal	federal	museum	policy	council
british	home	press	affairs	fund
california	world	international (61)	defense	act
Loc^0	Loc^1	Loc^2	Loc^3	Loc^4
washington (92)	the	texas	county	monday
los angeles	st.	new york	city	thursday
south	new	washington (22)	beach	river (57)
north	national (69)	united states	valley	tuesday
old	east (65)	baltimore	island	wednesday
grand	mount	california	river (71)	hotel
black	fort	capitol	park	friday
west (22)	west (56)	christmas	bay	hall
east (21)	lake	bosnia	house	center
haiti	great	san juan	supreme court	building

Table 3: 10 most common words for each grammar symbol. Words which appear in multiple places have observed counts indicated in parentheses.

Pers-gov	Pers-mod	Org-gov	Org-mod	Loc-gov	Loc-mod
according-to (1044)	director	president-of	$	university-of	calif.
played-by	spokesman	chairman-of	giant	city-of	newspap[er]
directed-by	leader	director-of	opposit[e]	from-to	state
led-by	presid[ent]	according-to (786)	group	town-of	downtown
meeting-with	attorney	professor-at	pp	state-of	n.y.
from-to	candid[ate]	head-of	compan[y]	center-in	warrant
met-with	lawyer	department-of	journal	out-of	va.
letter-to	chairman	member-of	firm	is-in	fla.
secretary-of	counsel	members-of	state	house-of	p.m.
known-as	actor	spokesman-for	agenc[y]	known-as	itself

Table 4: 10 most common prepositional and modifier features for each named entity class. Modifiers were Porter-stemmed; for clarity a reconstructed stem is shown in brackets.

on dev, 25% people). When the pronoun features are added, our system's performance increases to 86%, significantly better than the EM system.

One motivation for our use of a structured model which defined a notion of consistency between entities was that it might allow the construction of an unsupervised alias detector. According to the model, two entities are consistent if they are in the same class, and do not have conflicting assignments of words to lexical symbols. Results here are at best equivocal. The model is reasonable at passing basic tests– "Dr. Seuss" is not consistent with "Dr. Strangelove", "Dr. Quinn" etc, despite their shared title, because the model identifies the second element of each as a last name. Also correctly, "Dr. William F. Gibson" is judged consistent with "Dr. Gibson" and "Gibson" despite the missing elements. But mistakes are commonplace. In the "Gibson" case, the string "William F." is misanalyzed as a multiword string, making the name inconsistent with "William Gibson"; this is probably the result of a search error, which, as we explained, Gibbs sampling is unlikely to correct. In other cases, the system clusters a family group together under a single "entity" nonterminal by forcing their first names into inappropriate states, for instance assigning $Pers^1$ Bruce, $Pers^2$ Ellen, $Pers^3$ Jarvis, where $Pers^2$ (usually a middle name) actually contains the first name of a different individual. To improve this aspect of our system, we might incorporate name-specific features into the prior, such as abbreviations and the concept of a family name. The most critical improvement, however, would be integration with a generative coreference system, since the document context probably provides hints about which entities are and are not coreferent.

The other key issue with our system is inference. Currently we are extremely vulnerable to falling into local minima, since the complex structure of the model can easily lock a small group of examples into a poor configuration. (The "William F. Gibson" case above seems to be one of these.) In addition to the block sampler used by Bhattacharya and Getoor (2006), we are investigating general-purpose split-merge samplers (Jain and Neal, 2000) and the permutation sampler (Liang et al., 2007a). One interesting question is how well these samplers perform when faced with thousands of clusters (entities).

Despite these issues, we clearly show that it is possible to build a good model of named entity class while retaining compatibility with generative systems and without supervision. In addition, we do a reasonable job learning the latent structure of names in each named entity class. Our system improves over the latent named-entity tagging in Haghighi and Klein (2007), from 61% to 87%. This suggests that it should indeed be possible to improve on their coreference results without using a supervised named-entity model. How much improvement is possible in practice, and whether joint inference can also improve named-entity performance, remain interesting questions for future work.

Acknowledgements

We thank three reviewers for their comments, and NSF for support via grants 0544127 and 0631667.

References

Indrajit Bhattacharya and Lise Getoor. 2006. A latent dirichlet model for unsupervised entity resolution. In *The SIAM International Conference on Data Mining (SIAM-SDM)*, Bethesda, MD, USA.

William J. Black, Fabio Rinaldi, and David Mowatt. 1998. Facile: Description of the ne system used for muc-7. In *In Proceedings of the 7th Message Understanding Conference*.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL-09)*, Athens, Greece.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *Proc. of the 2005 Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 173–180.

Eugene Charniak. 2001. Unsupervised learning of name structure from coreference data. In *NAACL-01*.

Nancy A. Chinchor. 1998. Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, page 21 pages, Fairfax, VA, April. version 3.5, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Michael Collins and Yorav Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of EMNLP 99*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, Prague, Czech Republic, June. Association for Computational Linguistics.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana maria Popescu, Tal Shaked, Stephen Soderl, Daniel S. Weld, and Er Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134.

Richard Evans. 2003. A framework for named entity recognition in the open domain. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2003)*, pages 137 – 144, Borovetz, Bulgaria, September.

Sharon Goldwater, Tom Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems (NIPS) 18*.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 848–855. Association for Computational Linguistics.

Sonia Jain and Radford M. Neal. 2000. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182.

Mark Johnson, Tom L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of NAACL 2007*.

Xin Li, Paul Morie, and Dan Roth. 2004. Identification and tracing of ambiguous names: Discriminative and generative approaches. In *AAAI*, pages 419–424.

Percy Liang, Michael I. Jordan, and Ben Taskar. 2007a. A permutation-augmented sampler for DP mixture models. In *Proceedings of ICML*, pages 545–552, New York, NY, USA. ACM.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007b. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of EMNLP-CoNLL*, pages 688–697, Prague, Czech Republic, June. Association for Computational Linguistics.

A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG System Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Fairfax, Virginia.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30(1).

Radford M. Neal. 2003. Slice sampling. *Annals of Statistics*, 31:705–767.

Vincent Ng. 2008. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 640–649, Honolulu, Hawaii, October. Association for Computational Linguistics.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145, New York, NY, USA. ACM.

Jim Pitman and Marc Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Ann. Probab.*, 25:855–900.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 472–479. AAAI.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Olga Uryupina. 2004. Evaluating name-matching for coreference resolution. In *Proceedings of LREC 04*, Lisbon.