Recall-Oriented Learning for Named Entity Recognition in Wikipedia

Behrang Mohit, Nathan Schneider, Rishav Bhowmick,
Kemal Oflazer, and Noah A. Smith

CMU-LTI-11-012

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Recall-Oriented Learning for Named Entity Recognition in Wikipedia

Behrang Mohit* **Nathan Schneider†** **Rishav Bhowmick*** **Kemal Oflazer*** **Noah A. Smith†**
School of Computer Science
Carnegie Mellon University
*Doha, Qatar †Pittsburgh, PA 15213, USA
{behrang@,nschneid@cs.,rishavb@qatar.,ko@cs.,nasmith@cs.}cmu.edu

Abstract

We consider the problem of NER in Arabic Wikipedia, a semi-supervised domain adaptation setting for which we have no labeled training data in the target domain. To facilitate evaluation, we obtain annotations for articles in four topical groups, allowing annotators to identify domain-specific entity types in addition to standard categories. Standard supervised learning on newswire text leads to poor target-domain recall. We train a sequence model and show that a simple modification to the online learner—a loss function encouraging it to "arrogantly" favor recall over precision—substantially improves recall and F_1. We then employ self-training on unlabeled target-domain data in order to adapt our model; enforcing the same recall-oriented bias in the self-training stage yields additional gains.

1 Introduction

This paper considers named entity recognition (NER) in text that is different from most past research on NER. Specifically, we consider Arabic Wikipedia articles with diverse topics beyond the commonly-used news domain. These data challenge past approaches in two ways:

First, Arabic is a morphologically rich language (Habash, 2010). Named entities are referenced using more complex syntactic constructions (cf. English, in which NEs are primarily sequences of proper nouns). The Arabic script suppresses most vowels, increasing lexical ambiguity, and lacks capitalization, a key clue for English NER.

Second, much research has focused on the use of news text for system building and evaluation. Wikipedia articles are not news, belonging instead to a wide range of domains that are not clearly delineated. One hallmark of this divergence between Wikipedia and the news domain is a difference in the distributions of named entities. Indeed, the classic named entity types (person, organization, location) may not be the most apt for articles in other domains (e.g., scientific or social topics). On the other hand, Wikipedia is a large dataset, inviting semi-supervised approaches.

In this paper, we describe advances on the problem of NER in Arabic Wikipedia. The techniques are general and make use of well-understood building blocks. Our contributions are:

- A small corpus of articles annotated in a new scheme that provides more freedom for annotators to adapt named entity analysis to new domains;
- An "arrogant" learning approach designed to boost recall in supervised training as well as self-training, and
- An empirical evaluation of this technique as applied to a well-established discriminative NER model and feature set.

Token 2-way agreement rate	92.6% Cohen's κ = 0.86
Token 8-way agreement rate	88.3% Cohen's κ = 0.86
Token F_1 between annotators	91.0%
Entity boundary match F_1	94.0%
Entity category match F_1	87.4%

Table 1: Inter-annotator agreement measurements.

Experiments show consistent gains on the challenging problem of identifying named entities in Arabic Wikipedia text.

2 Arabic Wikipedia NE Annotation

Most of the effort in NER has been focused around a small set of domains and general-purpose entity classes relevant to those domains—especially the categories PER(SON), ORG(ANIZATION), and LOC(ATION) (POL), which are highly prominent in news text. Arabic is no exception: the publicly available NER corpora—ACE (Walker et al., 2006), ANER (Benajiba et al., 2008), and OntoNotes (Hovy et al., 2006)—all are within the news domain.[1] However, entity classes most relevant to the text will vary widely by domain; occurrence rates for entity classes are quite different in news text vs. Wikipedia, for instance (Balasuriya et al., 2009). This is abundantly clear in technical and scientific discourse, where much of the terminology is domain-specific, but it extends to other topics and genres as well. Non-POL entities that stand out in the history domain name, e.g., important events (wars, famines); cultural movements (*romanticism*); and political, religious, scientific, and literary texts. Ignoring such domain-critical entities in some sense misses the point.

Recognizing this limitation, some work on NER has sought to codify more robust inventories of general-purpose entity types (Sekine et al., 2002; Weischedel and Brunstein, 2005; Grouin et al., 2011) or to enumerate domain-specific types (Settles, 2004; Yao et al., 2003). Coarse, general-purpose categories have also been used for semantic tagging of nouns and verbs (Ciaramita and Johnson, 2003). Yet as the number of classes or domains grows, rigorously documenting and organizing the classes—even for a single language—requires intensive effort. Ideally, an NER system would identify new entity classes when they arise in new domains, adapting to new data. For this reason, we believe it is valuable to consider NER systems that identify (but do not label) entity mentions, and also to consider annotation schemes that allow annotators more freedom in defining entity classes. To that end, we have developed a small corpus of Arabic Wikipedia articles annotated for named entities.

Our aim in creating an annotated dataset is to provide a testbed for *evaluation* of new NER models. We will use these data as development and testing examples, but not as training data. In section 4 we will discuss our semi-supervised approach to learning, which leverages ACE and ANER data as an annotated training corpus.

2.1 Annotation Strategy

We conducted a small annotation project on Arabic Wikipedia articles. Two college-educated native Arabic speakers annotated about 3,000 sentences from 31 articles. We identified four topical areas of interest—history, technology, science, and sports—and browsed these topics until we had found 31 articles that we deemed satisfactory on the basis of length (at least 1,000 words), cross-lingual linkages (associated articles

[1] OntoNotes is only news-related text. ACE includes some text from blogs. In addition to the standard POL classes, both corpora include additional NE classes such as facility, event, product, vehicle, etc. However these entities are neither frequent nor comprehensive enough for covering the larger set of possible NEs (Sekine et al., 2002). Nezda et al. (2006) annotated and evaluated an Arabic NE corpus with an extended set of 18 classes (including temporal and numeric entities); this corpus has not been distributed publicly.

	History	Science	Sports	Technology
dev	Damascus	Atom	Raúl Gonzáles	Linux
	Imam Hussein Shrine	Nuclear power	Real Madrid	Solaris
test	Crusades	Enrico Fermi	2004 Summer Olympics	Computer
	Islamic Golden Age	Light	Christiano Ronaldo	Computer Software
	Islamic History	Periodic Table	Football	Internet
	Ibn Tolun Mosque	Physics	Portugal football team	Richard Stallman
	Ummaya Mosque	Muhammad al-Razi	FIFA World Cup	X Window System

Claudio Filippone (PER) كلوديو فيلبون — Linux (SOFTWARE) لينكس

Spanish League (CHAMPIONSHIPS) الدوري الاسباني — proton (PARTICLE) بروتون

nuclear radiation (GENERIC-MISC) الاشعاع النووي — Real Zaragoza (ORG) ريال سرقسطة

Table 2: Translated titles of Arabic Wikipedia articles in our development and test sets, and some NEs with standard and article-specific classes. Additionally, Prussia and Amman were reserved for training annotators, and Gulf War for estimating inter-annotator agreement.

in English, German, and Chinese[2]), and subjective judgments of quality. The list of these articles along with sample NEs are presented in table 2. These articles were then preprocessed to extract main article text (eliminating tables, lists, info-boxes, captions, etc.) for annotation.

Our approach follows ACE guidelines (LDC, 2005) in identifying NE boundaries and choosing POL tags. In addition to this traditional form of annotation, annotators were encouraged to articulate one to three *salient, article-specific* entity categories. For example, names of particles (e.g., *proton*) are highly salient in the Atom article. Annotators were asked to read the entire article first, and then to decide which non-traditional classes of entities would be important in the context of article. In some cases, annotators reported using heuristics (such as being proper nouns or having an English translation which is conventionally capitalized) to help guide their determination of non-canonical entities and entity classes. Annotators produced written descriptions of their classes, including example instances. For the purpose of this paper, we consider all article salient NEs and other infrequent NEs as being labeled as the *miscellaneous class* or MIS.

This scheme was chosen for its flexibility: in contrast to a scenario with a fixed ontology, annotators required minimal training beyond the POL conventions, and did not have to worry about delineating custom categories precisely enough that they would extend straightforwardly to other topics or domains. Of course, we expect inter-annotator variability to be greater for these open-ended classification criteria.

Below, we aim to develop entity detection models that generalize beyond the traditional POL entities. We leave to future work the challenges of automatically *classifying* entities into non-canonical types and inferring relationships among these classes. Hereafter, we merge all article-specific categories with the generic miscellaneous category.

2.2 Annotation Quality Evaluation

During annotation, two articles (Prussia and Amman) were reserved for training annotators on the nature of the task. Once they were accustomed to annotation, both independently annotated a third article. We used this 4,750-word article (Gulf War, حرب الخليج الثانية) to measure inter-annotator agreement. Table 1 provides scores for token-level agreement measures,[3] as well as entity-level F_1 between the two annotated versions of the article.

[2]These three languages have the most articles on Wikipedia. Associated articles here are those that have been manually hyperlinked from the Arabic page as cross-lingual correspondences. They are not translations, but if the associations are accurate, these articles should be topically similar to the Arabic page that links to them.

[3]To avoid artificial inflation of the agreement rate, we exclude the 81% of tokens tagged by both annotators as not belonging to an entity. As there are four classes (POLM), there are $|\{B, I\}| \times 4 = 8$ possible token-level tags. "2-way" agreement is between B and I only.

History: "Gulf War," "Prussia," "Damascus," "Crusades"		WAR_CONFLICT	● ● ●
Science: "Atom," "Periodic table"			
	THEORY ●	CHEMICAL	● ●
	NAME_ROMAN ●	PARTICLE	● ●
Sports: "Football," "Raúl Gonzáles"			
	SPORT ○	CHAMPIONSHIP	●
	AWARD ○	NAME_ROMAN	●
Technology: "Computer," "Richard Stallman"			
COMPUTER_VARIETY ○		SOFTWARE	●
		COMPONENT	●

Table 3: Custom NE categories suggested by one or both annotators for 10 articles. Article titles are translated from Arabic. ● indicates that both annotators volunteered a category for an article; ○ indicates that only one annotator suggested the category. Annotators were not given a predetermined set of possible categories; rather, category matches between annotators were determined by post hoc analysis. NAME_ROMAN indicates an NE rendered in Roman characters.

These measures indicate strong agreement for locating and categorizing NEs both at the token and chunk levels. Closer examination of agreement scores shows that PER and MIS classes have the lowest rates of agreement. That the miscellaneous class, used for infrequent or article-specific NEs, receives poor agreement is unsurprising. The low agreement on the PER class seems to be due to the use of titles and descriptive terms in personal names. Despite explicit guidelines to exclude the titles, annotators disagreed on the inclusion of descriptors that disambiguate the NE (e.g., *the father* in جرج بوش الآب: George Bush, the father).

2.3 Validating Category Intuitions

To investigate the variability between annotators with respect to custom category intuitions, we asked our two annotators to independently read 10 of the articles in the data (scattered across our four focus domains) and suggest up to 3 custom categories for each. One annotator suggested 14 categories; the other suggested 15. We assigned short names to these suggestions, seen in table 3. In 13 cases, both annotators suggested a category for an article that was essentially the same (●); three such categories spanned multiple articles. In three cases a category was suggested by only one annotator (○).[4] Thus, we see that our annotators were generally, but not entirely, consistent with each other in their creation of custom categories. Further, we observe that almost all of our article-specific categories correspond to classes in the extended NE taxonomy of (Sekine et al., 2002), which speaks to the reasonableness of both sets of categories—and by extension, our open-ended annotation process.

Our annotation of named entities outside of the traditional POL classes creates a useful resource for entity detection and recognition in new domains. Even the ability to detect non-canonical types of NEs should help applications such as QA and MT (Toral et al., 2005; Babych and Hartley, 2003). Possible avenues for future work include annotating and projecting non-canonical NEs from English articles to their Arabic counterparts (cf. Shah et al., 2010), automatically clustering non-canonical types of entities into article-specific or cross-article classes (cf. Frietag, 2004), or using non-canonical classes to improve (author-specified) article categories in Wikipedia.

[4]When it came to tagging NEs, one of the two annotators was assigned to each article. Custom categories only suggested by the other annotator were ignored.

		documents	words	sents.	entities	MIS rate
Training	ACE+ANER	—	212,839	7,053	15,796	7%
	Wikipedia (unlabeled)	397	1,110,546	40,001	—	—
Development	ACE	—	7,776	250	638	3%
	Wikipedia (4 domains)	8	21,203	711	2,073	53%
Test	ACE	—	7,789	266	621	2%
	Wikipedia (4 domains)	20	52,650	1,976	3,781	37%
	history	5	13,046	381	1,158	6%
	science	5	15,151	667	882	61%
	sports	5	11,240	376	932	12%
	technology	5	13,213	552	809	83%

Table 4: Train and test corpora statistics for the ACE, ANER and Wikipedia articles. The last column records the proportion of annotated entities belonging to the miscellaneous category.

3 Data

Table 4 summarizes the various corpora used in this work. Our NE-annotated Wikipedia subcorpus, described above, consists of several Arabic Wikipedia articles from four focus domains.[5] We do not use these for supervised training data; they serve only as development data (8 articles) and test data (20 articles). A larger set of 397 Arabic Wikipedia articles, selected on the basis of quality heuristics, serves as unlabeled data for semi-supervised learning.

Our out-of-domain labeled NE data is drawn from the ANER (Benajiba et al., 2007) and the ACE-2005 (Walker et al., 2006) newswire corpora. Entity types in this data are POL categories (PER, ORG, LOC) and MIS. 10,000 words from the ACE corpus were held out as development and test data; the remainder of the ACE corpus and the entire ANER corpus form a 200,000-word corpus of labeled training data.

4 Models

Our starting point for statistical NER is a linear model over sequences, trained using the structured perceptron (Collins, 2002). This framework enables us to manipulate two key elements of the model: the features and the loss function used in training. It is closely related to the preponderance of recent research on statistical NER.

4.1 Linear Feature-Based Model

Let $\boldsymbol{x} = \langle x_1, \ldots, x_M \rangle$ denote an input sequence, here a sentence in Arabic, and $\boldsymbol{y} = \langle y_1, \ldots, y_M \rangle$ denote a sequence of tags that encode named entity boundaries and labels ("BIO" tags, denoting tokens that are at the beginning, inside or outside of a NE). Given input $\boldsymbol{x}$, a linear model chooses an output

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) \tag{1}$$

where $\mathbf{g}$ maps the input-output pair into $\mathbb{R}^D$, a D-dimensional feature space, and $\mathbf{w}$ is a weight vector in $\mathbb{R}^D$ that parameterizes the model. Equation 1 is known as the *decoding* problem.

In most NLP research with sequence models, $\mathbf{g}$ is designed to *factor* into local parts:

$$\mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{m=1}^{M+1} \mathbf{f}(\boldsymbol{x}, y_{m-1}, y_m) \tag{2}$$

[5] We downloaded a snapshot of Arabic Wikipedia (`http://ar.wikipedia.org/`) as of August 29, 2009, and preprocessed the articles to extract main body text and metadata using the `mwlib` package for Python (PediaPress, 2010).

Input: data $\langle\langle \boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)}\rangle\rangle_{m=1}^{M}$; number of iterations T; rate schedule $\langle\alpha^{(t)}\rangle_{t=1}^{T}$
Output: $\mathbf{w}$
$\mathbf{w} \leftarrow \mathbf{0}$
for $t = 1$ **to** T **do**
  choose $\langle \boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}\rangle$ u.a.r. from the data $\hat{\boldsymbol{y}} \leftarrow \arg\max_{\boldsymbol{y}} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}^{(t)}, \boldsymbol{y})$ (eq. 1)
  if $\hat{\boldsymbol{y}} \neq \boldsymbol{y}^{(t)}$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{(t)}(\mathbf{g}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) - \mathbf{g}(\boldsymbol{x}^{(t)}, \hat{\boldsymbol{y}}))$
  end
end

Algorithm 1: Training with the perceptron.

This enables the use of a familiar dynamic programming technique—the Viterbi algorithm—for exactly solving the decoding problem in equation 1. The restriction is that each feature may only depend on two adjacent word-labels at a time. Such a model makes similar independence assumptions to those of a hidden Markov model.

4.2 Learning and the Perceptron

The perceptron can be understood in two ways: (i) as a simple iterative algorithm for finding a hyperplane (in D-dimensional feature space) that separates correct $\boldsymbol{y}$ values from incorrect ones, shown as Algorithm 1,[6] or (ii) as an empirical risk minimizer. Though the first view is more common, we adopt the second to help elucidate our new learning algorithm.

Given training examples $\langle\langle \boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}\rangle, \ldots, \langle \boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)}\rangle\rangle$, empirical risk minimization seeks:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{n=1}^{N} \ell(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}, \mathbf{w}) \tag{3}$$

where ℓ is some loss function, usually convex in $\mathbf{w}$, that penalizes mistakes.[7] The perceptron's familiar online updates (innermost line of Algorithm 1) can be understood as stochastic subgradient ascent on equation 3, with the perceptron's loss function:

$$\ell_{perceptron}(\boldsymbol{x}, \boldsymbol{y}, \mathbf{w}) = \max_{\boldsymbol{y}'} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}') - \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) \tag{4}$$

Alternative learning approaches use different loss functions (e.g., CRFs use the log loss). Structured SVMs (Tsochantaridis et al., 2004), notably, incorporate the notion of *cost* or error into the loss function. Let $c(\boldsymbol{y}, \boldsymbol{y}')$ denote a measure of error when $\boldsymbol{y}$ is the correct answer but $\boldsymbol{y}'$ is predicted. The structured hinge loss is $\ell_{hinge}(\boldsymbol{x}, \boldsymbol{y}, \mathbf{w}) =$

$$\max_{\boldsymbol{y}'} \left(\mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}') + c(\boldsymbol{y}, \boldsymbol{y}')\right) - \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) \tag{5}$$

The maximization problem inside the parentheses is known as *cost-augmented decoding*. If c factors similarly to equation 2, then we can increase penalties for $\boldsymbol{y}$ that have more local mistakes. This raises the learner's awareness about how it will be evaluated. Incorporating cost-augmented decoding into the perceptron is not a new idea (Gimpel and Smith, 2010), and it relates closely to well-known "aggressive" algorithms for online max-margin learning (Crammer et al., 2006).

In NER, this extension can allow biasing the learner away from false negatives or false positives, in an unbalanced way (Gimpel and Smith, 2010). Gimpel and Smith define word-local cost functions that

[6] The perceptron is guaranteed to eventually find such a hyperplane if one exists (Collins, 2002).

[7] In most machine learning approaches, a regularization term is added to avoid overfitting, e.g., $\|\mathbf{w}\|_2^2$. The perceptron does not explicitly regularize, relying instead on the stochasticity in the online updates and averaging or voting at the end of learning to avoid overfitting. We use averaging for our experiments.

differently penalize precision errors (i.e., $y_m^{(n)} = O \wedge \hat{y}_m \neq O$ for the m^{th} word of example n) and recall errors (i.e., $y_m^{(n)} \neq O \wedge \hat{y}_m = O$). A key problem in semi-supervised learning for NER is poor *recall*, so we will penalize recall errors:

$$c(\boldsymbol{y}, \boldsymbol{y}') = \sum_{m=1}^{M} \begin{cases} \beta & \text{if } y_m \neq O \wedge y'_m = O \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

for a penalty parameter $\beta > 0$. We call our learner the "recall-oriented" perceptron (ROP), though more precisely it is stochastic subgradient ascent with the hinge loss (eq. 5) and the cost function in eq. 6. The change to Algorithm 1 is simply to use

$$\hat{\boldsymbol{y}} \leftarrow \arg\max_{\boldsymbol{y}'} \left(\mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}') + c(\boldsymbol{y}, \boldsymbol{y}') \right) \tag{7}$$

which can be accomplished with a slight change to the Viterbi algorithm.

We note that Minkov et al. (2006) similarly explored the recall vs. precision tradeoff in NER. Their technique was to directly tune the weight of a single feature—the feature marking O (non-entity tokens); a lower weight for this feature will incur a greater penalty for predicting O. Below we demonstrate that our method, which is less coarse, is more successful in our setting.[8]

In our experiments we will show that injecting "arrogance" into the learner via the recall-oriented loss function substantially improves recall, especially for non-POL entities (section 6.3).

4.3 Self-Training and Semi-Supervised Learning

As we will show experimentally, the differences between news text and Wikipedia text call for domain adaptation techniques. In the case of Arabic Wikipedia, there is no available labeled training data. Yet the available *unlabeled* data is vast; as such, Wikipedia would seem like an ideal candidate for semi-supervised learning.

Here we adapt self-training, a simple technique that leverages a supervised learner (like the perceptron) to perform semi-supervised learning (Clark et al., 2003; Mihalcea, 2004; McClosky et al., 2006). In our version, a model is trained on the labeled data, then used to label the unlabeled target data. We then iterate between training on the hypothetically-labeled target data (ignoring the original labeled set) and relabeling. See Algorithm 2. Before self-training we remove sentences hypothesized not to contain any named entity mentions. Our intention is to avoid further encouragement of the model toward low recall and we validate its effectiveness on the development set.

Section 6.2 presents experiments with the standard self-training approach. In section 6.3 we investigate the effect of integrating the recall-oriented perceptron described above, as opposed to a standard perceptron learner, within self-training.

4.4 Summary of the Approach

Our baseline approach is to use standard NER features, training using the perceptron (Algorithm 1) on ACE training data. An alternative that we consider is the use of the recall-oriented perceptron, trained on ACE data; this will obtain superior performance. Our main approach to *adapt* those models for Arabic Wikipedia, is self-training (Algorithm 2), for which we find the recall-oriented perceptron (playing the role of the inner supervised learner) produces a small gain in performance.

[8] The crucial distinction between the techniques is that our cost function adjusts the *whole* model in order to generally perform better at recall than precision on the training data.

Input: labeled data $\langle\langle \boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)}\rangle\rangle_{m=1}^{M}$; unlabeled data $\langle \bar{\boldsymbol{x}}^{(j)}\rangle_{j=1}^{J}$; supervised learner L; number of iterations T'
Output: $\mathbf{w}$
$\mathbf{w} \leftarrow L(\langle\langle \boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)}\rangle\rangle_{m=1}^{M})$
for $t = 1$ **to** T' **do**
  for $j = 1$ **to** J **do**
    $\hat{\boldsymbol{y}}^{(j)} \leftarrow \arg\max_{\boldsymbol{y}} \mathbf{w}^{\top}\mathbf{g}(\bar{\boldsymbol{x}}^{(j)}, \boldsymbol{y})$
  end
  $\mathbf{w} \leftarrow L(\langle\langle \bar{x}^{(j)}, \hat{\boldsymbol{y}}^{(j)}\rangle\rangle_{j=1}^{J})$
end

Algorithm 2: Self-training.

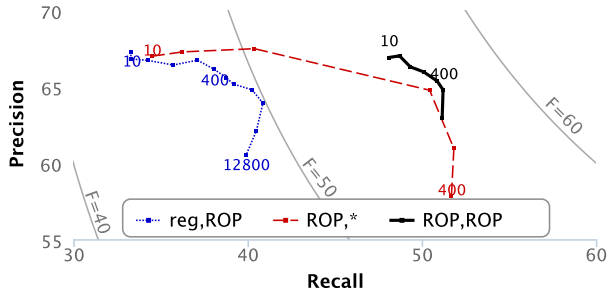


Figure 1: Tuning the recall-oriented cost parameter for different learning settings. We optimized for F_1, choosing penalty $\beta = 100$ for recall-oriented supervised learning (in the plot, ROP,*—this is regardless of whether a stage of self-training will follow); $\beta = 200$ for recall-oriented self-training following recall-oriented supervised learning (ROP,ROP); and $\beta = 3200$ for recall-oriented self-training following regular supervised learning (reg,ROP).

5 Features

Our features include many that have been found to work well for Arabic in past research and some new features enabled by Wikipedia. We do not make use of any gazetteer, viewing the construction of a broad-domain gazetteer as a significant undertaking on its own and orthogonal to the challenges of a new text domain like Wikipedia.

We use a first-order structured perceptron; none of our features consider more than a pair of BIO labels $\langle y_{m-1}, y_m\rangle$ at a time. The structured model enforces the constraint that NE sequences must begin with B (so the bigram $\langle O, I\rangle$ is disallowed).

Standard features. We include 15 contextual and lexical features capturing local context and shallow morphology. These consider a window of two previous words. They capture recurring parts of names, such as titles (e.g., *Dr.*) and descriptive nouns (e.g., *City* in *City of Cairo*). We also use the current word's length as some of the NEs have longer than average length. Following Abdul-Hamid and Darwish (2010), we extract a set of 12 character n-gram features: for a word w of length n, we extract leading and trailing unigrams, bigrams, and trigrams starting from the first and the last two letters of the word (e.g., $\mathbf{w}_1^2$, $\mathbf{w}_2^3$, $\mathbf{w}_{n-2}^{n-1}$, $\mathbf{w}_{n-1}^{n}$ for bigrams). The character-level features capture prefixes and suffixes that typify names, particularly names transliterated into Arabic from other languages (such as *-man* in German surnames). Interior n-grams can match similar affixes occurring inside clitics such as the conjunction *wa-*, the definite article *Al-*, and the plural suffix *-At* (as our tokenizer does not separate these).

Morphology and shallow syntax. Arabic words generally carry rich morphological information, some of which (including noun-adjective agreement and special markings for construct-state nominals in compounds) is local to noun phrases such as NEs. For this reason, morphological features have been found to be

useful in NE detection. Following Benajiba et al. (2008), we use the MADA toolkit (Habash and Rambow, 2005; Roth et al., 2008) to extract features encoding the normalized spelling, part-of-speech, aspect, case, number, gender, person, and definiteness/state of a word and its predecessor. As MADA was trained to achieve high performance on news text from the Arabic Treebank (Maamouri et al., 2004), we expect it to achieve somewhat lower accuracy on Wikipedia text. A small-scale evaluation of MADA's performance for Wikipedia data was encouraging, however; with regard to segmentation and diacriticization, less than 10% of MADA's output was judged faulty by both of our annotators.

Diacritics Though most words in Arabic text are unvocalized, diacritics are commonly used to disambiguate names the first time they appear in an article. This feature indicates whether the current word had diacritics in the source text. (Diacritics are removed from the token for all other purposes.)

Projected English capitalization. As has been noted previously (Benajiba et al., 2008), capitalization is an extremely useful cue for NER in English; Arabic is at a disadvantage in this regard because the script does not specially mark proper names. To correct this we turn to lexical correspondences between Arabic and English. One feature in our model indicates whether the word corresponds to a capitalized English word in MADA's Arabic-English lexicon. To improve recall in the Wikipedia domain, we construct a mapping between English and Arabic Wikipedia titles connected by cross-lingual links in article metadata.[9] Three indicator features encode whether the current word, the previous word, or the combination of the two map to a capitalized English term in this Wikipedia-derived lexicon.

6 Experiments

We investigate two questions in the context of NER for Arabic Wikipedia:

- **Loss function:** Does integrating a cost function into our learning algorithm, as we have done in the **recall-oriented perceptron** (section 4.2), improve recall and overall performance on Wikipedia data?
- **Semi-supervised learning for domain adaptation:** Can our models benefit from large amounts of unlabeled Wikipedia data, in addition to the (out-of-domain) labeled data? We experiment with a self-training phase following the fully supervised learning phase.

We report experiments for the possible combinations of the above ideas. These are summarized in table 5. Note that the recall-oriented perceptron can be used for the supervised learning phase, for the self-training phase, or both. This leaves us with the following combinations:

- **reg,none** (baseline): regular supervised learning only.
- **ROP,none**: recall-oriented supervised learner only.
- **reg,reg**: standard self-training setup.
- **ROP,reg**: recall-oriented supervised learner, followed by standard self-training.
- **reg,ROP**: regular supervised model as the initial labeler for recall-oriented self-training.
- **ROP,ROP** (the "double ROP" condition): recall-oriented supervised model as the initial labeler for recall-oriented self-training. Note that the two ROPs used here differ with respect to their cost parameter.

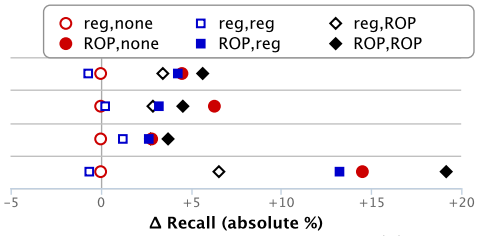
For evaluating our models we focus on the named entity *detection* task, i.e. recognizing which spans of words constitute entities. This is measured by per-entity precision, recall, and F_1.[10] All non-POL classes

[9] From full snapshots of the Arabic and English versions of Wikipedia, we collect titles of 85,642 Arabic articles doubly cross-linked to an English article (i.e., the Arabic article has a cross-lingual link to the English article, and vice versa). As the first letter of the title itself is automatically capitalized, we decide whether to capitalize the English side of the entry based on a heuristic measure of how consistently the title term is capitalized within the article. (Specifically, we threshold on the ratio of capitalized to uncapitalized occurrences of the title term within the English article.) We chose this method for its simplicity; others have developed more sophisticated techniques, namely gazetteers constructed in a semi-automated fashion from Wikipedia and other resources (Benajiba et al., 2008; Shaalan and Raza, 2008; Attia et al., 2010).

[10] Only entity spans that exactly match the gold spans are counted as correct. We calculated these scores with the `conlleval.pl` script from the CoNLL 2003 shared task.

SUPERVISED	SELF-TRAINING								
	none			**reg**			**ROP**		
reg	66.3	35.9	46.59	66.7	35.6	46.41	59.2	40.3	47.97
ROP	61.9	43.8	51.33	61.8	43.0	50.75	59.5	46.0	**51.88**

Table 5: Entity detection precision, recall, and F_1 for each learning setting, microaveraged across the 24 articles in our Wikipedia test set. Rows differ in the supervised learning condition on the ACE+ANER data (regular vs. recall-oriented perceptron). Columns indicate whether this supervised learning phase was followed by self-training on unlabeled Wikipedia data, and if so which version of the perceptron was used for self-training.



	entities	**words**	**baseline recall**
PER	1081	1743	49.95
ORG	286	637	23.92
LOC	1019	1413	61.43
MIS	1395	2176	9.30
overall	3781	5969	35.91

Figure 2: Recall improvement over baseline in the test set by gold NER category, counts for those categories in the data, and recall scores for our baseline model. Markers in the plot indicate different experimental settings corresponding to cells in table 5.

(including article-specific custom categories in the Wikipedia data) are collapsed into a single category, MIS. To measure statistical significance of differences between models we use Gimpel and Smith's (2010) implementation of the paired bootstrap resampler of (Koehn, 2004), taking 10,000 samples for each comparison.[11]

6.1 Baseline

Our baseline is the perceptron, trained on the POL entity boundaries in the ACE+ANER corpus.[12] Development data was used to select the number of iterations $T = 1$. We performed 3-fold cross-validation on the ACE data and found wide variance in the entity detection performance of this model:

	P	**R**	**F**
fold 1	70.43	63.08	66.55
fold 2	87.48	81.13	84.18
fold 3	65.09	51.13	57.27
average	74.33	65.11	69.33

(Fold 1 corresponds to the ACE test set described in table 4.) We also trained the model to perform POL detection and classification, achieving nearly identical results in the 3-way cross-validation of ACE data. From these data we conclude that our baseline is on par with the state of the art for Arabic NER on ACE news text (Abdul-Hamid and Darwish, 2010).[13]

Here is the performance of the baseline entity detection model on our 20-article Wikipedia test set:[14]

	P	**R**	**F**
technology	60.42	20.26	30.35
science	64.96	25.73	36.86
history	63.09	35.58	45.50
sports	71.66	59.94	65.28
overall	66.30	35.91	46.59

[11] Ordering the models by test set F_1, we find that all pairs of consecutive models are significantly different ($p < 0.05$), with the exception of the first two (regular supervised learning, regular vs. no self-training).

[12] In keeping with prior work, we ignore non-POL categories for the ACE evaluation.

[13] Abdul-Hamid and Darwish report as their best result a macroaveraged F-score of 76. Because they do not specify which data they used for their held-out test set, we cannot perform a direct comparison. However, our feature set is nearly a superset of their best feature set, and their result lies well within the range of results seen in our cross-validation folds.

[14] Our Wikipedia evaluations use models trained on POLM entity boundaries in ACE. Per-domain and overall scores are microaverages across articles.

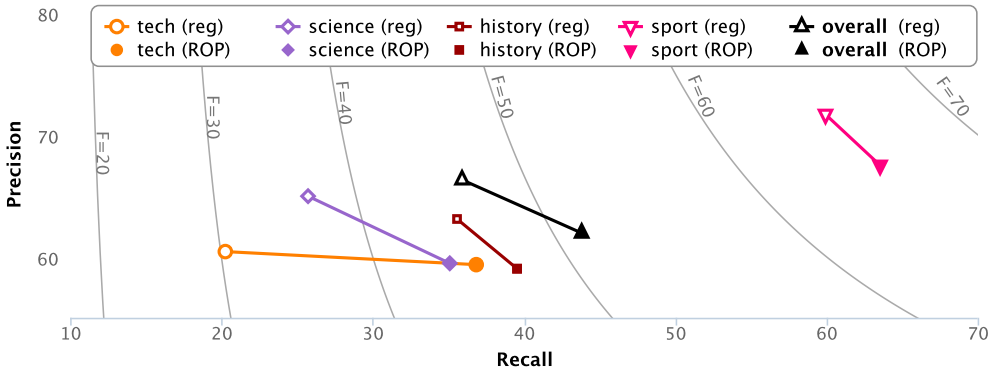


Figure 3: Supervised learner precision vs. recall as evaluated on Wikipedia test data in different topical domains. The regular perceptron (baseline model) is contrasted with ROP. No self-training is applied.

Unsurprisingly, performance on Wikipedia data varies widely across article domains and is much lower than in-domain performance. Though precision scores fall between 60% and 72% for all domains, recall in most cases is far worse. Miscellaneous class recall, in particular, suffers badly, weighing in at under 10%—which partially accounts for the poor recall in science and technology articles (those have by far the highest proportion of MIS entities; see table 4.) Thus, we explore methodologies to combat this recall deficit.

6.2 Self-Training

Following Clark et al. (2003), we applied self-training as described in Algorithm 2, with the perceptron as the supervised learner. Our unlabeled data consists of 397 Arabic Wikipedia articles (1 million words) selected at random from all articles exceeding a simple length threshold (1000 words); see table 4. We used only one iteration ($T' = 1$), as experiments on development data showed no benefit from additional rounds. Indeed, several rounds of self-training were found to hurt performance, an effect attested in much earlier research (Curran et al., 2007) and sometimes known as "semantic drift."

Results are shown in table 5; the middle column indicates the use of regular self-training. We find that standard self-training has very little impact on performance.[15] Why is this the case? We venture that poor baseline recall and the domain variability *within* Wikipedia are to blame. Limiting the unlabeled data to topics that are highly similar to the target topics and using new types of features/wider context could prove useful in this regard.

6.3 Recall-Oriented Learning

The recall-oriented bias can be introduced in either or both of the stages of our semi-supervised learning framework: in the supervised learning phase, modifying the objective of our baseline (section 6.1); and within the self-training algorithm (section 6.2).[16] As noted in section 4.2, the aim of this approach is to discourage recall errors (false negatives), which are the chief difficulty for the news text–trained model in the new domain. We selected the value of the false positive penalty used in cost-augmented decoding, β, using the development data (figure 1).

The results in table 5 demonstrate improvements due to the recall-oriented bias in both stages of learning. When used in the supervised phase (last row of the table), the recall gains are substantial—nearly 8% over the baseline. Integrating this bias within self-training (last column of the table) produces a more modest improvement of about 4% relative to the baseline. In both cases, the improvements to recall more than compensate for the amount of degradation to precision. This trend is robust: wherever the recall-oriented perceptron is added, we observe substantial gains in both recall and F_1.[17]

[15] In both settings, regular self-training produces a worse F_1 score than no self-training, though this is only significant when ROP supervised learning is used ($p < 0.05$).

[16] Standard Viterbi decoding was used to *label* the data within the self-training algorithm; note that cost-augmented decoding only makes sense in learning, not as a prediction technique, since it deliberately introduces errors.

[17] The worst of the three models with the ROP supervised learner is significantly better than the best of the models with the regular

The gains, perhaps surprisingly, are somewhat additive: using the ROP in both phases of learning is a significant (though small) improvement over alternatives (standard supervised perceptron, no self-training, or self-training with a standard perceptron). In fact, recall-oriented self-training succeeds despite the ineffectiveness of standard self-training.

Performance breakdowns by (gold) class, figure 2, and domain, figure 3, further attest to the robustness of the overall results. The most dramatic gains are in miscellaneous class recall—each form of the recall bias produces an improvement, and using this bias in both the supervised and self-training phases is clearly most successful for miscellaneous entities. Correspondingly, the technology and science domains (in which this class dominates) receive the biggest boost. Still, the gaps between domains are not entirely removed; the sports domain remains the indisputable winner.

An alternative—and even simpler—approach to controlling the precision-recall tradeoff is the Minkov et al. (2006) strategy of tuning a single feature weight subsequent to learning (see section 4.2 above). We performed an oracle experiment to determine how this compares to recall-oriented learning in our setting. An oracle trained with the method of Minkov et al. outperforms the three models in table 5 that use the regular perceptron for the supervised phase of learning, but underperforms the supervised ROP conditions.[18]

Overall, we find that incorporating the recall-oriented bias in learning is fruitful for adapting to Wikipedia because the gains in recall outpace the damage to precision.

7 Discussion

To our knowledge, this work is the first suggestion that modifying the supervised learning criterion in a resource-rich domain, prior to application in a new domain, might reap benefits. Others have looked at regularization (Chelba and Acero, 2006) and feature design (Daumé III, 2007); we alter the loss function. Not surprisingly, the double-ROP approach harms performance on the original domain (on ACE data, we achieve 55.41% F_1, far below the standard perceptron). It may be that, even before a learner is presented with data in a new domain, models can be prepared for adaptation, sacrificing performance in the original domain.

The recall-oriented bias is not merely encouraging the learner to identify entities it has already seen in training. As recall increases, so does the number of distinct entities correctly detected that were not present in the training data: of the 2070 NE types in the test data that were never seen in training, only 450 were ever found by the baseline, versus 588 in the reg,ROP condition, 632 in the ROP,none condition, and 717 in the double-ROP condition.

We note finally that our method is a simple extension to the standard structured perceptron; cost-augmented inference is no more expensive than maximum inference, and the algorithmic change is equivalent to adding one additional feature. Our recall-oriented cost function is parameterized by a single value, β; recall is highly sensitive to the choice of this value (figure 1), and thus we anticipate that tuning on development data will, in general, be essential to leveraging the benefits of arrogance.

8 Related Work

Our approach draws on insights from work in the areas of NER, domain adaptation, NLP with Wikipedia, and semi-supervised learning. As all are broad areas of research, we highlight only the most relevant contributions here.

Research in Arabic NER has been focused on compiling and optimizing the gazetteers and feature sets for standard sequential modeling algorithms (Benajiba et al., 2008; Farber et al., 2008; Shaalan and Raza,

supervised learner ($p < 0.005$). Whichever supervised learner is used, ROP self-training is significantly superior to alternatives ($p < 0.05$; $p < 0.005$ vs. regular self-training).

[18] Tuning the O feature weight to optimize for F_1 on our test set, we found that oracle precision would be 66.2, recall would be 39.0, and F_1 would be 49.1. The F_1 score of our best model is nearly 3 points higher than the Minkov et al.–style oracle, and over 4 points higher than the non-oracle version where the development set is used for tuning.

2008; Abdul-Hamid and Darwish, 2010). We make use of features identified in this prior work to construct a strong baseline system. We are unaware of any Arabic NER work that has addressed diverse text domains like Wikipedia. Both the English and Arabic versions of Wikipedia have been used, however, as resources in service of traditional NER (Kazama and Torisawa, 2007; Benajiba et al., 2008). Attia et al. (2010) heuristically induce a mapping between Arabic Wikipedia and Arabic WordNet to construct Arabic NE gazetteers.

Balasuriya et al. (2009) highlight the substantial divergence between entities appearing in English Wikipedia versus traditional corpora, and the effects of this divergence on NER performance. There is evidence that models trained on Wikipedia data generalize and perform well on corpora with narrower domains. Nothman et al. (2009) and Balasuriya et al. (2009) show that NER models trained on both automatically and manually annotated Wikipedia corpora perform reasonably well on news corpora. The reverse scenario does not hold for models trained on news text, a result we also observe in Arabic NER. Other work has gone beyond the entity detection problem: Florian et al. (2004) additionally predict within-document entity coreference for Arabic, Chinese, and English ACE text, while Cucerzan (2007) aims to resolve every mention detected in English Wikipedia pages to a canonical article devoted to the entity in question.

The domain and topic diversity of NEs has been studied in the framework of domain adaptation research. A group of these methods use self-training and select the most informative features and training instances to adapt a source domain learner to the new target domain. Wu et al. (2009) bootstrap the NER leaner with a subset of unlabeled instances that bridge the source and target domains. Jiang and Zhai (2006) and Daumé III (2007) make use of some labeled target-domain data, augmenting the feature space of the source model with features specific to the target domain. Here, in contrast, we do not assume that any labeled data is available in the target domain. Another important distinction is that domain variation in this prior work is restricted to topically-related corpora (e.g. newswire vs. broadcast news), whereas in our work, there are major topical differences between the training and test corpora and consequently between the salient NE classes. In these respects our NER setting is closer to that of Florian et al. (2010), who recognize English entities in noisy text—including non-English news and heavily abbreviated financial transaction reports—via an ensemble of statistical taggers trained with different data and feature sets in order to accommodate different noise levels. Related to NER, template-based information extraction systems have been adapted to fill target domain–specific templates (Surdeanu et al., 2011) using manually-developed rules and gazetteers.

Self-training (Clark et al., 2003; Mihalcea, 2004; McClosky et al., 2006) is widely used in NLP and has inspired related techniques that learn from automatically labeled data (Liang et al., 2008; Petrov et al., 2010). Our self-training procedure differs from that of Mihalcea (2004) in that we use all of the automatically labeled examples, rather than filtering them based on a confidence score.

Cost functions have been used in non-structured classification settings to penalize certain types of errors more than others (Chan and Stolfo, 1998; Domingos, 1999; Kiddon and Brun, 2011). The goal of optimizing our structured NER model for recall is quite similar to the scenario explored by Minkov et al. (2006). To trade off between precision and recall, they propose tuning the weight for the feature corresponding to the O tag ("not part of an entity"). Our approach of incorporating such a bias in the loss function instead (section 4.2) is more effective (section 6.3).

9 Conclusion

We have considered the problem of learning an NER model suited to domains for which no labeled training data is available. A loss function to encourage recall over precision during supervised discriminative learning substantially improves recall and overall entity detection performance, especially when combined with a semi-supervised learning regimen incorporating the same bias. We have also developed a small corpus of Arabic Wikipedia articles via a flexible entity annotation scheme which extends beyond the traditional fixed-category approach to better represent the domain at hand.

Acknowledgments

We would like to thank Mariem Fekih Zguir and Reham Al Tamime for their assistance with annotation, Michael Heilman for providing his perceptron tagger implementation, and Nizar Habash and colleagues for providing the MADA toolkit. We also thank members of the ARK group at CMU, Hal Daumé, and anonymous reviewers for their valuable suggestions. This research has been supported by Qatar National Research Fund (QNRF) grant NPRP-08-485-1-083.

References

Ahmed Abdul-Hamid and Kareem Darwish. 2010. Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115, Uppsala, Sweden, July. Association for Computational Linguistics.

Mohammed Attia, Antonio Toral, Lamia Tounsi, Monica Monachini, and Josef van Genabith. 2010. An automatically built named entity lexicon for Arabic. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools*, EAMT '03.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 10–18, Suntec, Singapore, August. Association for Computational Linguistics.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. ANERsys: an Arabic named entity recognition system based on maximum entropy. In Alexander Gelbukh, editor, *Proceedings of CICLing*, pages 143–153, Mexico City, Mexio. Springer.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2008. Arabic named entity recognition using optimized feature sets. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 284–293, Honolulu, Hawaii, October. Association for Computational Linguistics.

Philip K. Chan and Salvatore J. Stolfo. 1998. Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 164–168, New York City, New York, USA, August. AAAI Press.

Ciprian Chelba and Alex Acero. 2006. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech and Language*, 20(4):382 – 399.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in WordNet. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 168–175.

Stephen Clark, James Curran, and Miles Osborne. 2003. Bootstrapping POS-taggers using unlabelled data. In Walter Daelemans and Miles Osborne, editors, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 49–55.

Michael Collins. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, December.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June.

James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with Mutual Exclusion Bootstrapping. In *Proceedings of PACLING, 2007*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June. Association for Computational Linguistics.

Pedro Domingos. 1999. MetaCost: a general method for making classifiers cost-sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164.

Benjamin Farber, Dayne Freitag, Nizar Habash, and Owen Rambow. 2008. Improving NER in Arabic using a morphological tagger. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 2509–2514, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, page 18, Boston, Massachusetts, USA, May. Association for Computational Linguistics.

Radu Florian, John Pitrelli, Salim Roukos, and Imed Zitouni. 2010. Improving mention detection robustness to noisy input. In *Proceedings of EMNLP 2010*, pages 335–345, Cambridge, MA, October. Association for Computational Linguistics.

Dayne Freitag. 2004. Trained named entity recognition using distributional clusters. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 262–269, Barcelona, Spain, July. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin training for structured log-linear models. Technical Report CMU-LTI-10-008, Carnegie Mellon University.

Cyril Grouin, Sophie Rosset, Pierre Zweigenbaum, Karn Fort, Olivier Galibert, and Ludovic Quintard. 2011. Proposal for an extension of traditional named entities: from guidelines to evaluation, an overview. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 92–100, Portland, Oregon, USA, June. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan and Claypool Publishers.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL (HLT-NAACL)*, pages 57–60, New York City, USA, June. Association for Computational Linguistics.

Jing Jiang and ChengXiang Zhai. 2006. Exploiting domain structure for named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL (HLT-NAACL)*, pages 74–81, New York City, USA, June. Association for Computational Linguistics.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic, June. Association for Computational Linguistics.

Chloe Kiddon and Yuriy Brun. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 89–94, Portland, Oregon, USA, June. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

LDC. 2005. ACE (Automatic Content Extraction) Arabic annotation guidelines for entities, version 5.3.3. Linguistic Data Consortium, Philadelphia.

Percy Liang, Hal Daumé III, and Dan Klein. 2008. Structure compilation: trading structure for features. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 592–599, Helsinki, Finland.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

Rada Mihalcea. 2004. Co-training and self-training for word sense disambiguation. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, Boston, Massachusetts, USA.

Einat Minkov, Richard Wang, Anthony Tomasic, and William Cohen. 2006. NER systems that suit user's preferences: adjusting the recall-precision trade-off for entity extraction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 93–96, New York City, USA, June. Association for Computational Linguistics.

Luke Nezda, Andrew Hickl, John Lehmann, and Sarmad Fayyaz. 2006. What in the world is a *Shahab*? Wide coverage named entity recognition for Arabic. In *Proccedings of LREC*, pages 41–46.

Joel Nothman, Tara Murphy, and James R. Curran. 2009. Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 612–620, Athens, Greece, March. Association for Computational Linguistics.

PediaPress. 2010. mwlib. http://code.pediapress.com/wiki/wiki/mwlib.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 705–713, Cambridge, MA, October. Association for Computational Linguistics.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT*, pages 117–120, Columbus, Ohio, June. Association for Computational Linguistics.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of LREC*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110, Geneva, Switzerland, August. COLING.

Khaled Shaalan and Hafsa Raza. 2008. Arabic named entity recognition from diverse text types. In *Advances in Natural Language Processing*, pages 440–451. Springer.

Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. 2010. SYNERGY: a named entity recognition system for resource-scarce languages such as Swahili using online machine translation. In Guy De Pauw, Handré Groenewald, and Gilles-Maurice de Schryver, editors, *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pages 21–26, Valletta, Malta. European Language Resources Association (ELRA).

Mihai Surdeanu, David McClosky, Mason R. Smith, Andrey Gusev, and Christopher D. Manning. 2011. Customizing an information extraction system to a new domain. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, Portland, Oregon, USA, June. Association for Computational Linguistics.

Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Muñoz. 2005. Improving question answering using named entity recognition. *Natural Language Processing and Information Systems*, 3513/2005:181–191.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In Russ Greiner and Dale Schuurmans, editors, *Proceedings of the 21st International Machine Learning Conference (ICML)*, Banff, Canada. ACM Press.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. LDC2006T06, Linguistic Data Consortium, Philadelphia.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. LDC2005T33, Linguistic Data Consortium, Philadelphia.

Dan Wu, Wee Sun Lee, Nan Ye, and Hai Leong Chieu. 2009. Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1523–1532, Singapore, August. Association for Computational Linguistics.

Tianfang Yao, Wei Ding, and Gregor Erbach. 2003. CHINERS: a Chinese named entity recognition system for the sports domain. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 55–62, Sapporo, Japan, July. Association for Computational Linguistics.