Design Challenges and Misconceptions in Named Entity Recognition*†‡

Lev Ratinov **Dan Roth**
Computer Science Department
University of Illinois
Urbana, IL 61801 USA
{ratinov2,danr}@uiuc.edu

Abstract

We analyze some of the fundamental design challenges and misconceptions that underlie the development of an efficient and robust NER system. In particular, we address issues such as the representation of text chunks, the inference approach needed to combine local NER decisions, the sources of prior knowledge and how to use them within an NER system. In the process of comparing several solutions to these challenges we reach some surprising conclusions, as well as develop an NER system that achieves 90.8 F_1 score on the CoNLL-2003 NER shared task, the best reported result for this dataset.

1 Introduction

Natural Language Processing applications are characterized by making complex interdependent decisions that require large amounts of prior knowledge. In this paper we investigate one such application– Named Entity Recognition (NER). Figure 1 illustrates the necessity of using prior knowledge and non-local decisions in NER. In the absence of mixed case information it is difficult to understand that *"BLINKER"* is a person. Likewise, it is not obvious that the last mention of *"Wednesday"* is an organization (in fact, the first mention of *"Wednesday"* can also be understood as a "comeback" which happens on Wednesday). An NER system could take advantage of the fact that "blinker" is also mentioned later in the text as the easily identifiable "Reggie Blinker". It is also useful to know that *Udinese* is a soccer club (an entry about this club appears in Wikipedia), and the expression *"both Wednesday and Udinese"* implies that *"Wednesday"* and *"Udinese"* should be assigned the same label.

SOCCER - [PER BLINKER] BAN LIFTED .
[LOC LONDON] 1996-12-06 [MISC Dutch] forward [PER Reggie Blinker] had his indefinite suspension lifted by [ORG FIFA] on Friday and was set to make his [ORG Sheffield Wednesday] comeback against [ORG Liverpool] on Saturday . [PER Blinker] missed his club's last two games after [ORG FIFA] slapped a worldwide ban on him for appearing to sign contracts for both [ORG Wednesday] and [ORG Udinese] while he was playing for [ORG Feyenoord].

Figure 1: Example illustrating challenges in NER.

The above discussion focuses on the need for external knowledge resources (for example, that *Udinese* can be a soccer club) and the need for non-local features to leverage the multiple occurrences of named entities in the text. While these two needs have motivated some of the research in NER in the last decade, several other fundamental decisions must be made. These include: what model to use for

* The system and the Webpages dataset are available at: http://l2r.cs.uiuc.edu/∼cogcomp/software.php

† This work was supported by NSF grant NSF SoD-HCER-0613885, by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC and by an NDIIPP project from the National Library of Congress.

‡ We thank Nicholas Rizzolo for the baseline LBJ NER system, Xavier Carreras for suggesting the word class models, and multiple reviewers for insightful comments.

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pages 147–155,
Boulder, Colorado, June 2009.

sequential inference, how to represent text chunks and what inference (decoding) algorithm to use.

Despite the recent progress in NER, the effort has been dispersed in several directions and there are no published attempts to compare or combine the recent advances, leading to some design misconceptions and less than optimal performance. In this paper we analyze some of the fundamental design challenges and misconceptions that underlie the development of an efficient and robust NER system. We find that BILOU representation of text chunks significantly outperforms the widely adopted BIO. Surprisingly, naive greedy inference performs comparably to beamsearch or Viterbi, while being considerably more computationally efficient. We analyze several approaches for modeling non-local dependencies proposed in the literature and find that none of them clearly outperforms the others across several datasets. However, as we show, these contributions are, to a large extent, independent and, as we show, the approaches can be used together to yield better results. Our experiments corroborate recently published results indicating that word class models learned on unlabeled text can significantly improve the performance of the system and can be an alternative to the traditional semi-supervised learning paradigm. Combining recent advances, we develop a publicly available NER system that achieves 90.8 F_1 score on the CoNLL-2003 NER shared task, the best reported result for this dataset. Our system is robust – it consistently outperforms all publicly available NER systems (e.g., the Stanford NER system) on all three datasets.

2 Datasets and Evaluation Methodology

NER system should be robust across multiple domains, as it is expected to be applied on a diverse set of documents: historical texts, news articles, patent applications, webpages etc. Therefore, we have considered three datasets: CoNLL03 shared task data, MUC7 data and a set of Webpages we have annotated manually. In the experiments throughout the paper, we test the ability of the tagger to adapt to new test domains. Throughout this work, we train on the CoNLL03 data and test on the other datasets *without retraining*. The differences in annotation schemes across datasets created evaluation challenges. We discuss the datasets and the evaluation methods below.

The CoNLL03 shared task data is a subset of Reuters 1996 news corpus annotated with 4 entity types: *PER,ORG, LOC, MISC*. It is important to notice that *both* the training and the development datasets are news feeds from *August* 1996, while the test set contains news feeds from *December* 1996. The named entities mentioned in the test dataset are considerably different from those that appear in the training or the development set. As a result, the test dataset is considerably harder than the development set. **Evaluation:** Following the convention, we report phrase-level F_1 score.

The MUC7 dataset is a subset of the North American News Text Corpora annotated with a wide variety of entities including people, locations, organizations, temporal events, monetary units, and so on. Since there was no direct mapping from temporal events, monetary units, and other entities from MUC7 and the MISC label in the CoNLL03 dataset, we measure performance only on *PER,ORG* and *LOC*. **Evaluation:** There are several sources of inconsistency in annotation between MUC7 and CoNLL03. For example, since the MUC7 dataset does not contain the *MISC* label, in the sentence *"balloon, called the Virgin Global Challenger"* , the expression *Virgin Global Challenger* should be labeled as *MISC* according to CoNLL03 guidelines. However, the gold annotation in MUC7 is *"balloon, called the* [*ORG Virgin*] *Global Challenger"*. These and other annotation inconsistencies have prompted us to relax the requirements of finding the exact phrase boundaries and measure performance using token-level F_1.

Webpages - we have assembled and manually annotated a collection of 20 webpages, including personal, academic and computer-science conference homepages. The dataset contains 783 entities (96-loc, 223-org, 276-per, 188-misc). **Evaluation:** The named entities in the webpages were highly ambiguous and very different from the named entities seen in the training data. For example, the data included sentences such as : *"Hear, O Israel, the Lord our God, the Lord is one."* We could not agree on whether *"O Israel"* should be labeled as *ORG, LOC,* or *PER*. Similarly, we could not agree on whether *"God"* and *"Lord"* is an *ORG* or *PER*. These issues

led us to report token-level entity-identification F_1 score for this dataset. That is, if a named entity token was identified as such, we counted it as a correct prediction ignoring the named entity type.

3 Design Challenges in NER

In this section we introduce the baseline NER system, and raise the fundamental questions underlying robust and efficient design. These questions define the outline of this paper. NER is typically viewed as a sequential prediction problem, the typical models include HMM (Rabiner, 1989), CRF (Lafferty et al., 2001), and sequential application of Perceptron or Winnow (Collins, 2002). That is, let $\mathbf{x} = (x_1, \ldots, x_N)$ be an input sequence and $\mathbf{y} = (y_1, \ldots, y_N)$ be the output sequence. The sequential prediction problem is to estimate the probabilities

$$P(y_i | x_{i-k} \ldots x_{i+l}, y_{i-m} \ldots y_{i-1}),$$

where k, l and m are small numbers to allow tractable inference and avoid overfitting. This conditional probability distribution is estimated in NER using the following baseline set of features (Zhang and Johnson, 2003): (1) previous two predictions y_{i-1} and y_{i-2} (2) current word x_i (3) x_i word type (all-capitalized, is-capitalized, all-digits, alphanumeric, etc.) (4) prefixes and suffixes of x_i (5) tokens in the window $c = (x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})$ (6) capitalization pattern in the window c (7) conjunction of c and y_{i-1}.

Most NER systems use additional features, such as POS tags, shallow parsing information and gazetteers. We discuss additional features in the following sections. We note that we normalize dates and numbers, that is *12/3/2008* becomes **Date**, *1980* becomes **DDDD** and *212-325-4751* becomes **DDD*-*DDD*-*DDDD**. This allows a degree of abstraction to years, phone numbers, etc.

Our baseline NER system uses a regularized averaged perceptron (Freund and Schapire, 1999). Systems based on perceptron have been shown to be competitive in NER and text chunking (Kazama and Torisawa, 2007b; Punyakanok and Roth, 2001; Carreras et al., 2003) We specify the model and the features with the LBJ (Rizzolo and Roth, 2007) modeling language. We now state the four fundamental design decisions in NER system which define the structure of this paper.

Algorithm	Baseline system	Final System
Greedy	83.29	90.57
Beam size=10	83.38	90.67
Beam size=100	83.38	90.67
Viterbi	83.71	N/A

Table 1: Phrase-level F_1 performance of different inference methods on CoNLL03 test data. Viterbi cannot be used in the end system due to non-local features.

Key design decisions in an NER system.

1) How to represent text chunks in NER system?

2) What inference algorithm to use?

3) How to model non-local dependencies?

4) How to use external knowledge resources in NER?

4 Inference & Chunk Representation

In this section we compare the performance of several inference (decoding) algorithms: greedy left-to-right decoding, Viterbi and beamsearch. It may appear that beamsearch or Viterbi will perform much better than naive greedy left-to-right decoding, which can be seen as beamsearch of size one. The Viterbi algorithm has the limitation that it does not allow incorporating some of the non-local features which will be discussed later, therefore, we cannot use it in our end system. However, it has the appealing quality of finding the most likely assignment to a second-order model, and since the baseline features only have second order dependencies, we have tested it on the baseline configuration.

Table 1 compares between the greedy decoding, beamsearch with varying beam size, and Viterbi, both for the system with baseline features and for the end system (to be presented later). Surprisingly, the greedy policy performs well, this phenmenon was also observed in the POS tagging task (Toutanova et al., 2003; Roth and Zelenko, 1998). The implications are subtle. First, due to the second-order of the model, the greedy decoding is over 100 times faster than Viterbi. The reason is that with the BILOU encoding of four NE types, each token can take 21 states (*O, B-PER, I-PER , U-PER, etc.*). To tag a token, the greedy policy requires 21 comparisons, while the Viterbi requires 21^3, and this analysis carries over to the number of classifier invocations. Furthermore, both beamsearch and Viterbi require transforming the predictions of the classi-

Rep. Scheme	CoNLL03 Test	CoNLL03 Dev	MUC7 Dev	MUC7 Test
BIO	89.15	**93.61**	86.76	85.15
BILOU	**90.57**	93.28	**88.09**	**85.62**

Table 2: End system performance with BILOU and BIO schemes. BILOU outperforms the more widely used BIO.

fiers to probabilities as discussed in (Niculescu-Mizil and Caruana, 2005), incurring additional time overhead. Second, this result reinforces the intuition that global inference over the second-order HMM features does not capture the non-local properties of the task. The reason is that the NEs tend to be short chunks separated by multiple "outside" tokens. This separation "breaks" the Viterbi decision process to independent maximization of assignment over short chunks, where the greedy policy performs well. On the other hand, dependencies between isolated named entity chunks have *longer*-range dependencies and are not captured by second-order transition features, therefore requiring separate mechanisms, which we discuss in Section 5.

Another important question that has been studied extensively in the context of shallow parsing and was somewhat overlooked in the NER literature is the representation of text segments (Veenstra, 1999). Related works include voting between several representation schemes (Shen and Sarkar, 2005), lexicalizing the schemes (Molina and Pla, 2002) and automatically searching for best encoding (Edward, 2007). However, we are not aware of similar work in the NER settings. Due to space limitations, we do not discuss all the representation schemes and combining predictions by voting. We focus instead on two most popular schemes– BIO and BILOU. The BIO scheme suggests to learn classifiers that identify the **B**eginning, the **I**nside and the **O**utside of the text segments. The BILOU scheme suggests to learn classifiers that identify the **B**eginning, the **I**nside and the **L**ast tokens of multi-token chunks as well as **U**nit-length chunks. The BILOU scheme allows to learn a more expressive model with only a small increase in the number of parameters to be learned. Table 2 compares the end system's performance with BIO and BILOU. Examining the results, we reach two conclusions: (1) choice of encoding scheme has a big impact on the system performance and (2) the less used BILOU formalism significantly outperforms the widely adopted BIO tagging scheme. We use the BILOU scheme throughout the paper.

5 Non-Local Features

The key intuition behind non-local features in NER has been that identical tokens should have identical label assignments. The sample text discussed in the introduction shows one such example, where all occurrences of *"blinker"* are assigned the *PER* label. However, in general, this is not always the case; for example we might see in the same document the word sequences *"Australia"* and *"The bank of Australia"*. The first instance should be labeled as *LOC*, and the second as *ORG*. We consider three approaches proposed in the literature in the following sections. Before continuing the discussion, we note that we found that adjacent documents in the CoNLL03 and the MUC7 datasets often discuss the same entities. Therefore, we ignore document boundaries and analyze global dependencies in 200 and 1000 token windows. These constants were selected by hand after trying a small number of values. We believe that this approach will also make our system more robust in cases when the document boundaries are not given.

5.1 Context aggregation

(Chieu and Ng, 2003) used features that aggregate, for each document, the context tokens appear in. Sample features are: *the longest capitilized sequence of words in the document which contains the current token* and *the token appears before a company marker such as ltd. elsewhere in text*. In this work, we call this type of features *context aggregation features*. Manually designed context aggregation features clearly have low coverage, therefore we used the following approach. Recall that for each token instance x_i, we use as features the tokens in the window of size two around it: $c_i = (x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})$. When the same token type t appears in several locations in the text, say $x_{i_1}, x_{i_2}, \ldots, x_{i_N}$, for each instance x_{i_j}, in addition to the context features c_{i_j}, we also aggregate the context across all instances within 200 tokens: $C = \cup_{j=1}^{j=N} c_{i_j}$.

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + Context Aggregation	85.40	89.99	**79.16**	71.53	70.76
3)	(1) + Extended Prediction History	**85.57**	**90.97**	78.56	**74.27**	72.19
4)	(1)+ Two-stage Prediction Aggregation	85.01	89.97	75.48	72.16	**72.72**
5)	All Non-local Features (1-4)	86.53	90.69	81.41	73.61	71.21

Table 3: The utility of non-local features. The system was trained on CoNLL03 data and tested on CoNNL03, MUC7 and Webpages. No single technique outperformed the rest on all domains. The combination of all techniques is the most robust.

5.2 Two-stage prediction aggregation

Context aggregation as done above can lead to excessive number of features. (Krishnan and Manning, 2006) used the intuition that some instances of a token appear in easily-identifiable contexts. Therefore they apply a baseline NER system, and use the resulting predictions as features in a second level of inference. We call the technique *two-stage prediction aggregation*. We implemented the token-majority and the entity-majority features discussed in (Krishnan and Manning, 2006); however, instead of document and corpus majority tags, we used relative frequency of the tags in a 1000 token window.

5.3 Extended prediction history

Both context aggregation and two-stage prediction aggregation treat all tokens in the text similarly. However, we observed that the named entities in the beginning of the documents tended to be more easily identifiable and matched gazetteers more often. This is due to the fact that when a named entity is introduced for the first time in text, a canonical name is used, while in the following discussion abbreviated mentions, pronouns, and other references are used. To break the symmetry, when using beamsearch or greedy left-to-right decoding, we use the fact that when we are making a prediction for token instance x_i, we have already made predictions $y_1, \ldots, y_{i-1}$ for token instances $x_1, \ldots, x_{i-1}$. When making the prediction for token instance x_i, we record the label assignment distribution for all token instances for the same token type in the previous 1000 words. That is, if the token instance is *"Australia"*, and in the previous 1000 tokens, the token type *"Australia"* was twice assigned the label *L-ORG* and three times the label *U-LOC*, then the prediction history feature will be: $(L-ORG:\frac{2}{5}; U-LOC:\frac{3}{5})$.

5.4 Utility of non-local features

Table 3 summarizes the results. Surprisingly, no single technique outperformed the others on all datasets. The extended prediction history method was the best on CoNLL03 data and MUC7 test set. Context aggregation was the best method for MUC7 development set and two-stage prediction was the best for Webpages. Non-local features proved less effective for MUC7 test set and the Webpages. Since the named entities in Webpages have less context, this result is expected for the Webpages. However, we are unsure why MUC7 test set benefits from non-local features much less than MUC7 development set. Our key conclusion is that no single approach is better than the rest and that the approaches are complimentary- their combination is the most stable and best performing.

6 External Knowledge

As we have illustrated in the introduction, NER is a knowledge-intensive task. In this section, we discuss two important knowledge resources– gazetteers and unlabeled text.

6.1 Unlabeled Text

Recent successful semi-supervised systems (Ando and Zhang, 2005; Suzuki and Isozaki, 2008) have illustrated that unlabeled text can be used to improve the performance of NER systems. In this work, we analyze a simple technique of using word clusters generated from unlabeled text, which has been shown to improve performance of dependency parsing (Koo et al., 2008), Chinese word segmentation (Liang, 2005) and NER (Miller et al., 2004). The technique is based on word class models, pioneered by (Brown et al., 1992), which hierarchically

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + Gazetteer Match	87.22	91.61	85.83	80.43	74.46
3)	(1) + Word Class Model	86.82	90.85	80.25	79.88	72.26
4)	All External Knowledge	88.55	92.49	84.50	83.23	74.44

Table 4: Utility of external knowledge. The system was trained on CoNLL03 data and tested on CoNNL03, MUC7 and Webpages.

clusters words, producing a binary tree as in Figure 2.

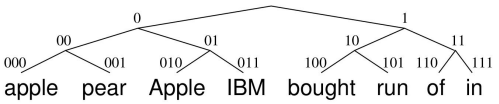

Figure 2: An extract from word cluster hierarchy.

The approach is related, but not identical, to distributional similarity (for details, see (Brown et al., 1992) and (Liang, 2005)). For example, since the words *Friday* and *Tuesday* appear in similar contexts, the Brown algorithm will assign them to the same cluster. Successful abstraction of both as a *day of the week*, addresses the data sparsity problem common in NLP tasks. In this work, we use the implementation and the clusters obtained in (Liang, 2005) from running the algorithm on the Reuters 1996 dataset, a superset of the CoNLL03 NER dataset. Within the binary tree produced by the algorithm, each word can be uniquely identified by its path from the root, and this path can be compactly represented with a bit string. Paths of different depths along the path from the root to the word provide different levels of word abstraction. For example, paths at depth 4 closely correspond to POS tags. Since word class models use large amounts of unlabeled data, they are essentially a semi-supervised technique, which we use to considerably improve the performance of our system.

In this work, we used path prefixes of length 4,6,10, and 20. When Brown clusters are used as features in the following sections, it implies that all features in the system which contain a word form will be duplicated and a new set of features containing the paths of varying length will be introduced. For example, if the system contains the feature *concatenation of the current token and the system prediction on the previous word*, four new features will be introduced which are concatenations of the previous prediction and the 4,6,10,20 length path-representations of the current word.

6.2 Gazetteers

An important question at the inception of the NER task was whether machine learning techniques are necessary at all, and whether simple dictionary lookup would be sufficient for good performance. Indeed, the baseline for the CoNLL03 shared task was essentially a dictionary lookup of the entities which appeared in the training data, and it achieves 71.91 F_1 score on the test set (Tjong and De Meulder, 2003). It turns out that while problems of coverage and ambiguity prevent straightforward lookup, injection of gazetteer matches as features in machine-learning based approaches is critical for good performance (Cohen, 2004; Kazama and Torisawa, 2007a; Toral and Munoz, 2006; Florian et al., 2003). Given these findings, several approaches have been proposed to automatically extract comprehensive gazetteers from the web and from large collections of unlabeled text (Etzioni et al., 2005; Riloff and Jones, 1999) with limited impact on NER. Recently, (Toral and Munoz, 2006; Kazama and Torisawa, 2007a) have successfully constructed high quality and high coverage gazetteers from Wikipedia.

In this work, we use a collection of 14 high-precision, low-recall lists extracted from the web that cover common names, countries, monetary units, temporal expressions, etc. While these gazetteers have excellent accuracy, they do not provide sufficient coverage. To further improve the coverage, we have extracted 16 gazetteers from Wikipedia, which collectively contain over 1.5M entities. Overall, we have 30 gazetteers (available for download with the system), and matches against

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + External Knowledge	88.55	92.49	84.50	83.23	74.44
3)	(1) + Non-local	86.53	90.69	81.41	73.61	71.21
4)	**All Features**	**90.57**	**93.50**	**89.19**	**86.15**	**74.53**
5)	All Features (train with dev)	90.80	N/A	89.19	86.15	74.33

Table 5: End system performance by component. Results confirm that NER is a knowledge-intensive task.

each one are weighted as a separate feature in the system (this allows us to trust each gazetteer to a different degree). We also note that we have developed a technique for injecting non-exact string matching to gazetteers, which has marginally improved the performance, but is not covered in the paper due to space limitations. In the rest of this section, we discuss the construction of gazetteers from Wikipedia.

Wikipedia is an open, collaborative encyclopedia with several attractive properties. (1) It is kept updated manually by it collaborators, hence new entities are constantly added to it. (2) Wikipedia contains redirection pages, mapping several variations of spelling of the same name to one canonical entry. For example, *Suker* is redirected to an entry about *Davor Šuker*, the Croatian footballer (3) The entries in Wikipedia are manually tagged with categories. For example, the entry about the *Microsoft* in Wikipedia has the following categories: *Companies listed on NASDAQ; Cloud computing vendors; etc.*

Both (Toral and Munoz, 2006) and (Kazama and Torisawa, 2007a) used the free-text description of the Wikipedia entity to reason about the entity type. We use a simpler method to extract high coverage and high quality gazetteers from Wikipedia. By inspection of the CoNLL03 shared task annotation guidelines and of the training set, we manually aggregated several categories into a higher-level concept (not necessarily NER type). When a Wikipedia entry was tagged by one of the categories in the table, it was added to the corresponding gazetteer.

6.3 Utility of External Knowledge

Table 4 summarizes the results of the techniques for injecting external knowledge. It is important to note that, although the world class model was learned on the superset of CoNLL03 data, and although the Wikipedia gazetteers were constructed based on CoNLL03 annotation guidelines, these features proved extremely good on all datasets. Word class models discussed in Section 6.1 are computed offline, are available online[1], and provide an alternative to traditional semi-supervised learning. It is important to note that the word class models and the gazetteers and independednt and accumulative. Furthermore, despite the number and the gigantic size of the extracted gazetteers, the gazeteers alone are not sufficient for adequate performance. When we modified the CoNLL03 baseline to include gazetteer matches, the performance went up from 71.91 to 82.3 on the CoNLL03 test set, below our baseline system's result of 83.65. When we have injected the gazetteers into our system, the performance went up to 87.22. Word class model and nonlocal features further improve the performance to 90.57 (see Table 5), by more than 3 F_1 points.

Dataset	Stanford-NER	LBJ-NER
MUC7 Test	80.62	85.71
MUC7 Dev	84.67	87.99
Webpages	72.50	74.89
Reuters2003 test	87.04	90.74
Reuters2003 dev	92.36	93.94

Table 6: Comparison: token-based F_1 score of LBJ-NER and Stanford NER tagger across several domains

7 Final System Performance Analysis

As a final experiment, we have trained our system both on the training and on the development set, which gave us our best F_1 score of 90.8 on the CoNLL03 data, yet it failed to improve the performance on other datasets. Table 5 summarizes the performance of the system.

Next, we have compared the performance of our

[1] http://people.csail.mit.edu/maestro/papers/bllip-clusters.gz

system to that of the Stanford NER tagger, across the datasets discussed above. We have chosen to compare against the Stanford tagger because to the best of our knowledge, it is the best publicly available system which is trained on the same data. We have downloaded the Stanford NER tagger and used the strongest provided model trained on the CoNLL03 data with distributional similarity features. The results we obtained on the CoNLL03 test set were consistent with what was reported in (Finkel et al., 2005). Our goal was to compare the performance of the taggers across several datasets. For the most realistic comparison, we have presented each system with a raw text, and relied on the system's sentence splitter and tokenizer. When evaluating the systems, we matched against the gold tokenization ignoring punctuation marks. Table 6 summarizes the results. Note that due to differences in sentence splitting, tokenization and evaluation, these results are not identical to those reported in Table 5. Also note that in this experiment we have used token-level accuracy on the CoNLL dataset as well. Finally, to complete the comparison to other systems, in Table 7 we summarize the best results reported for the CoNLL03 dataset in literature.

	System	Resources Used	F_1
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02
-	(Krishnan and Manning, 2006)	Non-local Features	87.24
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17
+	(Finkel et al., 2005)	Non-local Features	86.86

Table 7: Results for CoNLL03 data reported in the literature. publicly available systems marked by +.

8 Conclusions

We have presented a simple model for NER that uses expressive features to achieve new state of the art performance on the Named Entity recognition task. We explored four fundamental design decisions: text chunks representation, inference algorithm, using non-local features and external knowledge. We showed that BILOU encoding scheme significantly outperforms BIO and that, surprisingly, a conditional model that does not take into account interactions at the output level performs comparably to beamsearch or Viterbi, while being considerably more efficient computationally. We analyzed several approaches for modeling non-local dependencies and found that none of them clearly outperforms the others across several datasets. Our experiments corroborate recently published results indicating that word class models learned on unlabeled text can be an alternative to the traditional semi-supervised learning paradigm. NER proves to be a knowledge-intensive task, and it was reassuring to observe that knowledge-driven techniques adapt well across several domains. We observed consistent performance gains across several domains, most interestingly in Webpages, where the named entities had less context and were different in nature from the named entities in the training set. Our system significantly outperforms the current state of the art and is available to download under a research license.

Apendix– wikipedia gazetters & categories

1)**People**: *people, births, deaths*. Extracts 494,699 Wikipedia titles and 382,336 redirect links. 2)**Organizations**: *cooperatives, federations, teams, clubs, departments, organizations, organisations, banks, legislatures, record labels, constructors, manufacturers, ministries, ministers, military units, military formations, universities, radio stations, newspapers, broadcasters, political parties, television networks, companies, businesses, agencies*. Extracts 124,403 titles and 130,588 redirects. 3)**Locations**: *airports, districts, regions, countries, areas, lakes, seas, oceans, towns, villages, parks, bays, bases, cities, landmarks, rivers, valleys, deserts, locations, places, neighborhoods*. Extracts 211,872 titles and 194,049 redirects. 4)**Named Objects**: *aircraft, spacecraft, tanks, rifles, weapons, ships, firearms, automobiles, computers, boats*. Extracts 28,739 titles and 31,389 redirects. 5)**Art Work**: *novels, books, paintings, operas, plays*. Extracts 39,800 titles and 34037 redirects. 6)**Films**: *films, telenovelas, shows, musicals*. Extracts 50,454 titles and 49,252 redirects. 7)**Songs**: *songs, singles, albums*. Extracts 109,645 titles and 67,473 redirects. 8)**Events**: *playoffs, championships, races, competitions, battles*. Extracts 20,176 titles and 15,182 redirects.

References

R. K. Ando and T. Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *ACL*.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

X. Carreras, L. Màrquez, and L. Padró. 2003. Learning a perceptron-based named entity chunker via online recognition feedback. In *CoNLL*.

H. Chieu and H. T. Ng. 2003. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL*.

W. W. Cohen. 2004. Exploiting dictionaries in named entity extraction: Combining semi-markov extraction processes and data integration methods. In *KDD*.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.

L. Edward. 2007. Finding good sequential model structures using output transformations. In *EMNLP)*.

O. Etzioni, M. J. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named entity recognition through classifier combination. In *CoNLL*.

Y. Freund and R. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

J. Kazama and K. Torisawa. 2007a. Exploiting wikipedia as external knowledge for named entity recognition. In *EMNLP*.

J. Kazama and K. Torisawa. 2007b. A new perceptron algorithm for sequence labeling with non-local features. In *EMNLP-CoNLL*.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *ACL*.

V. Krishnan and C. D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *ACL*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*. Morgan Kaufmann.

P. Liang. 2005. Semi-supervised learning for natural language. *Masters thesis, Massachusetts Institute of Technology*.

S. Miller, J. Guinness, and A. Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL*.

A. Molina and F. Pla. 2002. Shallow parsing using specialized hmms. *The Journal of Machine Learning Research*, 2:595–613.

A. Niculescu-Mizil and R. Caruana. 2005. Predicting good probabilities with supervised learning. In *ICML*.

V. Punyakanok and D. Roth. 2001. The use of classifiers in sequential inference. In *NIPS*.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *IEEE*.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI*.

N. Rizzolo and D. Roth. 2007. Modeling discriminative global inference. In *ICSC*.

D. Roth and D. Zelenko. 1998. Part of speech tagging using a network of linear separators. In *COLING-ACL*.

H. Shen and A. Sarkar. 2005. Voting between multiple data representations for text chunking. *Advances in Artificial Intelligence*, pages 389–400.

J. Suzuki and H. Isozaki. 2008. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *ACL*.

E. Tjong, K. and F. De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

A. Toral and R. Munoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *EACL*.

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*.

J. Veenstra. 1999. Representing text chunks. In *EACL*.

T. Zhang and D. Johnson. 2003. A robust risk minimization based named entity recognition system. In *CoNLL*.