

## Named Entity Recognition by Neural Sliding Window

Ignazio Gallo, Elisabetta Binaghi and Moreno Carullo  
Università degli Studi dell'Insubria  
Dipartimento di Informatica e Comunicazione  
via Mazzini, 5 - 21100 Varese, Italy  
ignazio.gallo@uninsubria.it

Nicola Lamberti  
7pixel s.r.l.  
via Copernico, 2 - 20082 Binasco, Italy  
nicola@trovaprezzi.it

### Abstract

*Named Entity Recognition (NER) is an important sub-task of document processing such as Information Extraction. This paper describes a NER algorithm which uses a Multi-Layer Perceptron (MLP) to find and classify entities in natural language text. In particular we use the MLP to implement a new supervised context-based NER approach called Sliding Window Neural (SWiN). The SWiN method is a good solution for domains where the documents are grammatically ill-formed and it is difficult to exploit the features derived from linguistic analysis. Experiments indicate good accuracy compared with traditional approaches and demonstrate the system's portability.*

### 1. Introduction

The increasing popularity of the Web determines a huge growth of sources of information that are the premise for efficient on-line services in several domains. Within e-commerce applications for example, *price comparison services* obtain prices from retailers and allow individuals to see and compare them for specific products.

In this context Information Extraction (IE) techniques play a fundamental role allowing the automatic extraction of relevant facts from large collections of unstructured, or semi structured documents. Name Entity Recognition (NER) is a sub-task of Information Extraction in which proper nouns and numerical information in a document are detected and classified into categories.

Work on NER has been heavily funded by the U.S. government, beginning with the MUC<sup>1</sup> Conferences and continuing with the ACE<sup>2</sup> project.

<sup>1</sup>Message Understanding Conference (MUC)  
[http://www.itl.nist.gov/iaui/894.02/related\\_projects/muc/](http://www.itl.nist.gov/iaui/894.02/related_projects/muc/)

<sup>2</sup>Automatic Content Extraction (ACE) Evaluation  
<http://www.itl.nist.gov/iaui/894.01/tests/ace/>

Name Entity Recognition has been a well-studied problem for formal text which is characterised by well-formed sentences having seldom grammatical and spelling error; in this kind of text there are only few non-word characters, which are easily confused when doing Name Entity Recognition.

In e-commerce applications such as price comparison services, the automatic recognition of entities like models, brands and prices is of great value for further document processing. However in this case, as in many other context related to Web applications, NER approaches have to cope with critical aspects related to grammatically ill-formed sentences, ambiguities in writing entities to be recognised (e.g. models name).

Several NER approaches were proposed basically falling in three main groups [11]: dictionary-based, rule-based, and machine-learning-based. The dictionary-based approach assumes the presence of a dictionary of names of target types and identifies names in text by using string-matching techniques; performances of this approach are strongly affected by the dictionary coverage. The rule-based approach typically uses hand-crafted linguistic rules (or patterns/templates) and seeks named entities by pattern-matching or template-filling. The problem is that good rules require hard work from domain experts, and are not easily adaptable to new domains. In recent years, supervised learning techniques have become dominant, with better performance and adaptability [16, 2, 1, 18, 14, 13, 4, 12]. To build an accurate model, they need a collection of documents marked up with ground truth and a set of local features. The discovery of good features for NER is a critical task, implying creativity and domain knowledge. Some examples of local features are: *is in all caps*, *is in a list of city names*, *is in bold font*, etc.

Most supervised learning approaches may rely on background knowledge for pattern generalisation. For example, RAPIER [3] imposes content constraints based on the WordNet semantic classes. SRV [8] involves several features to map text tokens into some discrete value, for in-

stance, token length, character type (numeric or alphabetic), part-of-speech, etc. Using background knowledge, learning systems can generalise either delimiter-based rules [16] or context-based rules.

In the present work we address the problem of recognising name entities in commercial on-line documents containing informal text. A salient aspect of our strategy is the use of supervised learning technique based on a Multi-Layer Perceptron (MLP) model which acts as a contextual classifier. The neural model, named *Sliding Window Neural* (SWiN), receives contextual information by means of a sliding window mechanism that moves across the input documents.

## 2. The Proposed Model

In the present work the NER problem is treated as a one-class classification problem. In contrast with conventional classification approaches for NER that attempt to distinguish among several types of entities and then conceive the solution as a multi-classifier, our approach attempts to classify only one type of entity distinguishing it from all others.

Given a collection of documents  $D$ , the SWiN method is organised in the following steps:

**Tokenization** For each document  $d \in D$  provide a transformation  $T$  splitting the text in the document into a list of tokens  $T(d) = d^T = \{t_1, t_2, \dots, t_M\}$ . The Tokenization process allows to characterise the set of named entities  $E^d$  to be recognised in each document as constituted by elements  $E_i^d$  defined as specific sequence of tokens; formally  $E^d = \{E_1^d, E_2^d, \dots, E_N^d\}$ , with  $E_i^d = \{t_{k_i+1}, \dots, t_{k_i+R_i}\}$ , where  $R_i$  is the length (number of tokens) of the named entity  $E_i^d$ .

**Feature Extraction** For each token  $t_i$  extract a signature denoted by  $\beta_i$ . The signature consists of an array of  $G$  feature values  $\beta_i = \{f_{i,1}, f_{i,2}, \dots, f_{i,G}\}$ .

**Sliding Window** Starting at the first token, a sliding window  $S_W$  of dimension  $W$ , is moved across the list of tokens  $d^T$  with step  $p$ ,  $p \geq 1$ .

**Neural Computation** For each position  $j$  of the sliding window,  $W$  signatures  $\{\beta_{j+1}, \dots, \beta_{j+W}\}$  are provided in input to the neural network.

**Training** Training samples are presented to the neural network having the following form  $(P_{in}, P_{out})$ .  $P_{in} = \{\beta_1, \dots, \beta_W\}$  constitutes the input pattern and  $P_{out} = \{o_1, \dots, o_W\}$  is the expected output. The  $o_i$  component associated with a token belonging to a given entity  $E_j^d$  is set to 1, the others to 0. The Algorithm 1 describes in detail the

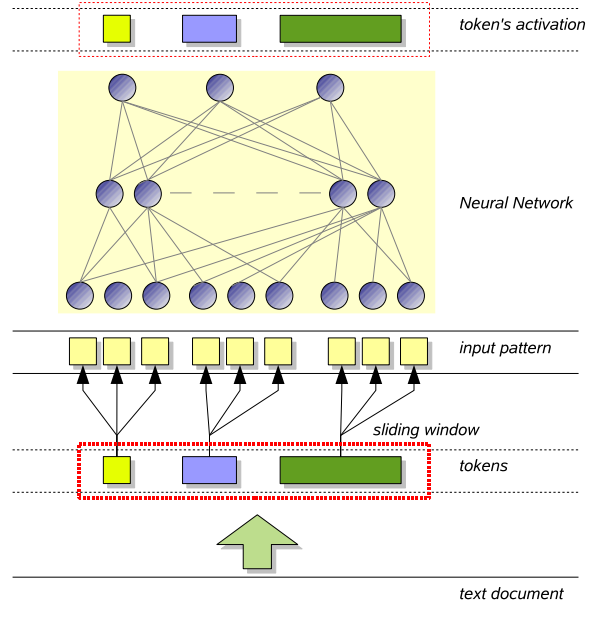


Figure 1. SWiN algorithm architecture

process used to extract the training set  $\mathbb{T}$  used for neural learning.

**Generalisation** The trained neural network performs recognition for never seen input patterns associating with each token  $t_i$  belonging to the input pattern an output value  $o_i$  indicating at which degree the token can be part of a given entity  $E_j^d$ . The generalisation process is described in detail in Algorithm 2.

During the generalisation phase, according to the step value  $p$  and size  $W$  of the Sliding Window  $S_W$ , each token can be classified more than once. A decision rule must be accomplished to compute the final degree of membership. In the present work for each token  $t_i$  the value of average activation  $\bar{o}_i = \frac{\sum_{n=1}^{N_i} o_{i,n}}{N_i}$  is calculated, where  $N_i$  is the number of times in which the token has been classified by the neural model; a token  $t_i$  belongs to the entity  $E_j^d$  if the average activation value  $\bar{o}_i$  exceeds a certain threshold  $\tau$ .

An overview of the proposed model is contained in Figure 1, where all the steps described above are depicted.

## 3. The Features

The extraction of an appropriate set of local features is a crucial task strongly affecting the performances of the overall NER strategy. The present work is aimed to preliminarily assess the feasibility and potentialities of the overall strategy without seeking the best features toward the best

---

**Algorithm 1** Training set extraction for the SWiN method

---

**Require:** Select the tokenizer  $T$

**Require:** Select the set of features

**Require:** Select the Sliding Window size  $W$

**Require:** Select the Sliding Window step  $p$

**Require:**  $\mathbb{T} \leftarrow \emptyset$

```
1: for all  $d \in D$  do
2:    $d^T \leftarrow T(d)$ 
3:    $j \leftarrow 0$ 
4:   repeat
5:      $S_W \leftarrow \{t_{j+1}, t_{j+2}, \dots, t_{j+W}\}$ 
6:      $P_{in} \leftarrow \bigcup_{z=j+1}^{j+W} \{f_{z,1}, f_{z,2}, \dots, f_{z,G}\}$ 
7:      $P_{out} \leftarrow \{o_{j+1}, o_{j+2}, \dots, o_{j+W}\}$ 
8:      $\mathbb{T} \leftarrow \mathbb{T} \cup (P_{in}, P_{out})$ 
9:      $j \leftarrow j + p$ 
10:  until  $(j + W + p) \leq M$ 
11: end for
```

---

---

**Algorithm 2** Generalisation using the trained neural model for the SWiN method

---

**Require:** Select the tokenizer  $T$ .

**Require:** Select the set of features .

**Require:** Load the trained neural network.

**Require:** Select the  $S_W$ 's size  $W$  according to the dimension of the input layer of the trained neural network.

**Require:** Set the Sliding Window step to  $p \leq W$ .

```
1: for all  $d \in D$  do
2:    $d^T \leftarrow T(d)$ 
3:    $j \leftarrow 0$ 
4:    $o_i \leftarrow 0, \forall t_i \in d^T$ 
5:   repeat
6:      $S_W \leftarrow \{t_{j+1}, t_{j+2}, \dots, t_{j+W}\}$ 
7:      $P_{in} \leftarrow \bigcup_{z=j+1}^{j+W} \{f_{z,1}, f_{z,2}, \dots, f_{z,G}\}$ 
8:     load  $P_{in}$  in input to the neural network and compute output values  $o_i$ 
9:      $\bar{o}_i = \bar{o}_i + o_i$  with  $i = j + 1, \dots, j + W$ 
10:     $j \leftarrow j + p$ 
11:   until  $(j + W + p) \leq M$ 
12:    $\bar{o}_i = \frac{\bar{o}_i}{N_i}$  with  $i = 1, \dots, M$ 
13: end for
```

---

performance. For this reason an initial minimal set of features is formulated specifically addressing non conventional domains in which documents characterised by ill-formed grammatical structure, make linguistic analysis inapplicable. The set of features we consider is the following:

- $f_1 = \mathbf{TokenType}$ : the token is transformed into a sequence of Boolean values each of which describes the token. For example, we can describe a token with the following set of Boolean values: (*FirstUpper*, *AllUpper*, *AllLower*, *Mixed*, *Number*), or (*areLetters*, *areDigits*, *arePunctuations*).
- $f_2 = \mathbf{KnownWords}$ : the token is transformed into a single Boolean value that represents the membership to a list of words that usually occur in documents related to a given Entity. Documents are preprocessed to compile a number of lists that are used by the SWiN algorithm.
- $f_3 = \mathbf{TokenLength}$ : The token is transformed into a single real value that describes its length. The real value is scaled in a range [0..1] by the following function

$$f(x) = \begin{cases} 0 & \text{se } x < 0 \\ \frac{x}{M} & \text{se } 0 \leq x \leq M \\ M & \text{se } x > M \end{cases} \quad (1)$$

where  $x = \text{length}(t_i)$  and  $M$  is a domain dependent constant.

- $f_4 = \mathbf{TokensDistance}$ : the token is transformed into a single real value that describes the distance (number of chars) between current token  $t_i$  and the previous  $t_{i-1}$ . The real value is scaled in a range [0..1] by the same function defined in equation (1).

## 4. Experiments

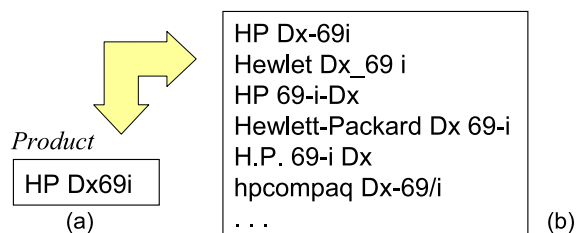
Two experiments were conceived and conducted. The first experiment is aimed to exploit the potentialities of SWiN strategy in dealing with short unstructured documents. In particular we considered the *commercial offers* domain<sup>3</sup>, where documents describe offers of commercial products made by on-line shopkeepers. In this domain, we selected six different product categories: *cell-phones*, *digital-cameras*, *monitor-lcd*, *notebook*, *PDA*<sup>4</sup>, *LCD-televisions*. The second experiment is based on a benchmark data set allowing a detailed comparison analysis. In particular we considered the well known *seminar*

<sup>3</sup>Document collection taken from the price comparison service Shoppydoo (<http://www.shoppydoo.it>), and available on <http://www.dicom.uninsubria.it/arteLab/ricerca.html>

<sup>4</sup>Personal Digital Assistant or PDA are also known as small computers or palmtop computers

**Table 1. Number of supervised documents contained in the six data-sets belonging to the commercial offers domain.**

Dataset	Train	Test
cell-phones	2557	1278
digital-cameras	5076	2537
monitor-lcd	2980	1490
notebook	4196	2098
PDA's	812	406
LCD-televitions	2862	1430



**Figure 2. List of potential descriptions in (b) that may be used to describe product (a).**

announcement v1.2 collection<sup>5</sup>, consisting of 485 postings to electronic bulletin boards that are part of the on-line environment of the Computer Science Department at Carnegie Mellon University.

#### 4.1. Evaluation Metrics

Much research in NER has been linked for years to the series of MUC conferences, that proposed evaluation metrics [5] such as precision (P) recall (R) and F-measure [7] ( $F_\beta$ ) described in the following.

Calling *response* an entity  $E_i^d$  returned by the system we can define the following concepts:

- *correct* = number of responses that are exactly what we expected (both boundaries are correct);
- *partial* = number of responses that are only partly what we expect to find (only one boundary is correct);
- *actual* = total number of responses from the system;
- *possible* = total number of responses that we expected from system;

<sup>5</sup>The data set was originally collected from newsgroups and annotated by Dayne Freitag and available on <http://www.isi.edu/info-agents/RISE/v1.2>, a corrected version of Seminar Announcements, is now available on <http://nlp.shef.ac.uk/dot.com/resources.html>

Using these definitions we define the measures used to evaluate the implemented system:

$$P = \frac{\text{correct} + (\text{partial} \times 0.5)}{\text{actual}} \quad (2)$$

$$R = \frac{\text{correct} + (\text{partial} \times 0.5)}{\text{possible}} \quad (3)$$

$$F_\beta = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R} \quad (4)$$

where the parameter  $\beta$  determines how much to favor recall over precision. It is typical for researchers in IE to report the F1 score of a system ( $\beta = 1$ ), which weights precision and recall equally [9].

#### 4.2. Experiments using Commercial Offers dataset

In the commercial offers collection, documents belonging to different categories are subdivided in training and testing data sets (Table 1). The purpose of every document in this collection is to describe a commercial product identified by model's name and brand. The named entity to be recognised in this collection is the name of the model described in the documents. In Table 2 an example for each category is provided; the text in bold is the model's name to be identified. Documents used in the experiment present various critical aspects for NER, essentially related to ambiguities in name formulation, ill-formed sentences, omissions and special characters. In Figure 2 different ways adopted by dealers to specify model names are exemplified.

The tokenizer decomposes input documents into sequences of *letters* and sequences of *digits*, ignoring any other character.

In a first experiment only the features  $f_1$  and  $f_2$  defined in section 3 are used extracting the following numerical values from each token:

- five boolean values for the feature  $f_1$ : *FirstUpper*, *AllUpper*, *AllLower*, *Mixed*, *Number*;
- two boolean values for feature  $f_2$ , representing the membership to two different lists of known words: *KnownModels* and *KnownBrands*. These two lists contain tokens extracted from all the names of models and names of brands contained in the training set.

The neural net was configured with a single hidden layer where the number of neurons correspond to those present in the input layer. The effect of the sliding window size on this dataset was studied varying the size of  $S_W$  from  $W = 7$  to

**Table 2. Examples of supervised documents contained in the six datasets belonging to the commercial offers domain.**

Category	Document example
cell-phones	NOKIA <b>1110i</b> blu ECCEZIONALE : Pagamento in contrassegno
digital-cameras	BenQ <b>DC E310</b> - Fotocamera digitale - 3.2 Mpix / 5 Mpix (interpolati) - memoria supportata: SD
monitor-lcd	MAXDATA 17 LCD <b>101728</b> RT 5ms Dual Input Ang.Vis. 160 Multim..SilverBlack TCO '03
notebook	ACER <b>TravelMate 4314WXM</b> i Sempron 3500+14+512+80+DL+X1100 256MB/512+2.2KG+Vista
PDA's	Mio Digiwalker Palmare Mio <b>P350</b> - Gps Integrato - No Sw
LCD-televisions	Hannspree - K206-10E1-010 Hannspree <b>Fantasy HANNSz.giraffe</b> - TV LCD - 9.6

$W = 9$ . Results in Table 3 show that a large window size give best performances.

A second group of experiments was conducted to study the generalisation capability of the proposed SWiN model. The model was first trained on a single category and then tested over all other categories. The  $S_W$  size used for these tests was  $W = 9$  while the features were  $f_1$  and  $f_2$  as described above.

Results are shown in Table 4; training on Cell-Phones category and testing on Digital-Cameras and Pads leads to discrete results while testing on other categories such as LCD-Televisions leads to bad results. Results obtained can be easily interpreted by considering examples in Table 2; good generalisation capabilities were reached for categories that show similarities in their description

### 4.3. Experiments using Seminar Announcement dataset

The seminar announcements is one of the most popular benchmark in the field of IE [3, 10, 17, 15, 6]. For this collection the goal is to extract four types of fragments from each article, those describing the start time (*stime*) and end time (*etime*) of the seminar, its location (*location*) and the name of the seminar speaker (*speaker*).

The data (485 documents) are randomly split into two sets of equal size, one for training and the other for testing.

The tokenizer  $T$ , used in this first group of experiments, decomposes each input document in  $D$  into two type of tokens  $t_i$ : sequences of *letters* and sequences of *digits*, ignoring any other character.

Here we used all the features defined in section 3. From each token were extracted the following numerical values:

- two boolean values for the feature  $f_1$ : *areLetters*, *areDigits*;

**Table 4. Generalisation test: F-measure (F1) obtained by a neural model trained on a single category of the commercial offers dataset and tested on all the six categories.**

	Tested on					
	cell-phones	digital-cameras	monitor-lcd	notebook	PDA's	LCD-televisions
Trained on						
cell-phones	<b>95.0</b>	71.5	42.8	43.4	67.6	29.7
digital-cameras	77.3	<b>92.2</b>	49.0	51.3	67.0	33.9
monitor-lcd	73.5	62.7	<b>78.4</b>	43.6	50.7	38.9
notebook	71.0	63.9	47.8	<b>84.9</b>	56.2	36.8
PDA's	76.0	68.2	34.8	41.0	<b>88.1</b>	21.5
LCD-televisions	49.6	38.2	39.9	35.8	35.0	<b>82.0</b>

**Table 3. Precision, Recall and F-measure obtained by SWiN varying the size of the window considering all the categories of the commercial offers dataset. In bold the best result in terms of F1 measure.**

category	win size = 7			win size = 8			win size = 9		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
cell-phones	95.3	95.1	<b>95.2</b>	95.0	94.2	94.6	95.2	94.8	95.0
digital-cameras	89.5	90.3	89.9	90.6	91.2	90.9	92.4	91.9	<b>92.2</b>
monitor-lcd	76.6	74.7	75.6	79.3	76.4	77.8	78.5	78.2	<b>78.4</b>
notebook	83.4	79.1	81.2	85.6	80.4	83.0	86.9	83.1	<b>84.9</b>
PDAs	89.2	89.2	89.2	86.6	86.9	86.7	88.0	88.2	<b>88.1</b>
LCD-televisions	81.1	75.9	78.4	81.8	80.4	81.1	83.5	80.7	<b>82.0</b>

- one boolean value for the feature  $f_2$ : the list of *Known-Words* was extracted from all the entities present in the training set;
- one real value scaled with  $M = 5$  referred to equation (1) for the feature  $f_3$ ;
- one real value scaled with  $M = 10$  referred to equation (1) for the feature  $f_4$ .

The  $S_W$  size was heuristically assessed with value  $W = 7$ ; the neural net was configured with a single hidden layer where the number of neurons correspond to those present in the input layer.

We report results using the same training/testing splits used by Freitag in original experiments with this corpus. The reported results are an average of ten runs. Table 5 presents a comparison with previous attempts at the CMU seminar corpus. Numerical results presented in this table were extracted from [15]. SWiN performs comparably to the best system in each category and in some cases it prevails. Looking into the details, considering the named entity *etime*, *location* and *speaker* best results are obtained in Recall, maintaining an high measure of Precision. It is worth to note that we used a minimal set of features without introducing additional optimised features.

## 5. Conclusions

Our objective in this work was to investigate the potentialities of a new contextual neural model for recognizing entities within informal documents. The model was tested on a standard data set for comparison and in a real e-commerce domain. As seen in the experimental context, the allied use of the supervised neural learning and the contextual feature extraction strategy benefits the NER process in general. In particular it allows to successfully deal with critical aspects arising from the informal structure of documents.

The experiment with standard dataset showed that SWiN reaches good accuracy compared with traditional approaches.

In the future work we want to test different neural models that are faster in training. Another critical task that we want to study is the selection of the optimal set of features in order to improve the results.

## References

- [1] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.
- [2] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *In Proc. of the Sixth ACL Workshop on Very Large Corpora*, 1998.
- [3] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the sixteenth national conference on Artificial intelligence (AAAI '99/IAAI '99)*, pages 328–334, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.
- [4] H. L. Chieu and H. T. Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 160–163, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
- [5] N. Chinchor. Muc-4 evaluation metrics. In *MUC4 '92: Proceedings of the 4th conference on Message understanding*, pages 22–29, Morristown, NJ, USA, 1992. Association for Computational Linguistics.
- [6] F. Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *Proc. of the 17th International Joint Conference On Artificial Intelligence (IJCAI2001)*, 2001.
- [7] W. B. Frakes and R. A. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.

**Table 5. Comparison using Seminar Announcement dataset, in terms of Precision, Recall and F-measure, between the proposed SWiN algorithm and numerous published results. In bold the best result comparing all the algorithms.**

System	stime			etime			loc			speaker		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
SNoW-IE	<b>99.6</b>	99.6	<b>99.6</b>	<b>97.6</b>	95	<b>96.3</b>	<b>90.9</b>	64.1	<b>75.2</b>	83.3	66.3	73.8
RAPIER-WT	96.5	95.3	95.9	94.9	94.4	94.6	91.0	61.5	73.4	79.0	40.0	53.1
RAPIER	93.9	92.9	93.4	95.8	94.6	95.2	91.0	60.5	72.7	80.9	39.4	53.0
SRV	98.6	98.4	98.5	67.3	92.6	77.9	74.5	70.1	72.2	54.4	58.4	56.3
WHISK	86.2	<b>100</b>	92.6	85.0	87.2	86.1	83.6	55.4	66.6	52.6	11.1	18.3
SWiN	91.0	91.2	91.4	92.2	<b>98.7</b>	95.3	74.8	<b>75.3</b>	75.1	<b>84.3</b>	<b>79.7</b>	<b>82.0</b>

- [8] D. Freitag. Information extraction from html: Application of a general machine learning approach. In *Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 517–523, 1998.
- [9] D. Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University, 1998.
- [10] D. Freitag and A. K. McCallum. Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.
- [11] B. Gu, V. Dahl, and F. Popowich. Recognizing biomedical named entities in the absence of human annotated corpora. In *Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2007)*.
- [12] I. Hendrickx and A. van den Bosch. Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data. In *Proceedings of CoNLL-2003*, pages 176–179, 2003.
- [13] H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
- [14] G. Paliouras, V. Karkaletsis, G. Petasis, and C. D. Spyropoulos. Learning decision trees for named-entity recognition and classification. In *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction*.
- [15] D. Roth and W. tau Yih. Relational learning via propositional algorithms: An information extraction case study. In *IJCAI*, pages 1257–1263, 2001.
- [16] S. Sekine, R. Grishman, and H. Shinnou. A decision tree method for finding and classifying names in japanese texts. In *WVLC-98 - Montreal, Canada*, 1998.
- [17] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.
- [18] G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of ACL'02*, pages 473–480, 2002.