Extracting person names from ancient Islamic Arabic texts

Majid Asgari Bidhendi, Behrouz Minaei-Bidgoli, Hosein Jouzi

School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
Computer Research Center of Islamic Sciences, Qom, Iran
majid_asgari@comp.iust.ac.ir, b_minaei@iust.ac.ir, hjuzi@noornet.net

Abstract

Recognizing and extracting name entities like person names, location names, date and time from an electronic text is very useful for text mining tasks. Named entity recognition is a vital requirement in resolving problems in modern fields like question answering, abstracting systems, information retrieval, information extraction, machine translation, video interpreting and semantic web searching. In recent years many researches in named entity recognition task have been lead to very good results in English and other European languages; whereas the results are not convincing in other languages like Arabic, Persian and many of South Asian languages. One of the most necessary and problematic subtasks of named entity recognition is person name extracting. In this article we have introduced a system for person name extraction in Arabic religious texts using proposed "Proper Name candidate injection" concept in a conditional random fields model. Also we have created a corpus from ancient Arabic religious texts. Experiments have shown that very hight efficient results have been obtained based on this approach.

Keywords: Arabic Named entity recognition, information extraction, conditional random fields

1. Introduction

Named entity identification that also known as Named entity recognition and name entity extraction, is a subtask of information extraction and that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations (companies, organizations and etc.), locations (cities, countries, rivers and etc.), time and dates, quantities, etc. As we have shown in next section, named entity extraction task and especially person name extracting haven't been lead to convincing results in Arabic language. Moreover, most of works which done for NER in Arabic language, have been focused on modern newspaper data. As we will show, there are very significant differences between newspaper data and ancient religious texts in Arabic language. In this paper we have focused specially on NER task for three main type of Islamic texts: historic, Hadith[1] and jurisprudential books. Person name extracting is very useful for Islamic religious sciences. Especially, for historic and Hadith books, finding the relationships between person names is a very valuable task. In Hadith books, people cited quotations from main religious leaders of Islam. These valuable data can help us to verify correctness of citations based on known truthful and untruthful relaters. Also we must point out that NER task is very useful subtask in text processing (and also in religious text processing) which can help other subtasks of natural language processing(Benajiba et al., 2004).

The rest of this paper is structured as follows. In section 2., we investigate the other efforts have been made to solve named entity task in Arabic language. In section 3., we emphasize on making a proper corpora for Arabic religious texts, NoorCorp, and NoorGazet, a gazetteer which contains religious person names. Section 4., explains in details Noor ANER system based on "proper name candidate injection" concept in conditional random fields model. Finally, in section 5.. we present the experiments we have carried out with our system, whereas in last section we draw our conclusions and future works.

2. Related works

Before 2008, the most successful documented and comparable efforts had been made in named entity recognition task was ANERsys (Benajiba et al., 2007) (based on maximum entropy), ANERsys 2 (Benajiba and Rosso, 2007) (based on conditional random fields) and proprietary Siraj system. In 2008, Benjiba and Rosso, proposed an another system based on combining results from four conditional random fields models (Benajiba and Rosso, 2008). Afterward Aljomaily et al. introduced an online system based on pattern recognition in (Al-Jumaily et al., 2011). Those patterns were built up by processing and integrating different gazetteers, from DBPedia (http://dbpedia.org/About, 2009) to GATE (A general architecture for text engineering, 2009) and ANERGazet (http://users.dsic.upv.es/grupos/nle/?file=kop4.php). All of these efforts have presented their results on ANERCorp that has been made by (Benajiba et al., 2007). The best results in extracting person named was obtained in (Al-Jumaily et al., 2011). They recorded F-measure equals to 76.28 for person names in their results. Furthermore some efforts have been made on NER task in specific domains. For example in (Fehri et al., 2011), F-measure equals to 94 has been obtained for sport news. In (Elsebai and Meziane, 2011), authors presented a method based on using a keyword set instead complicated syntactic, statistical or machine learning approaches.

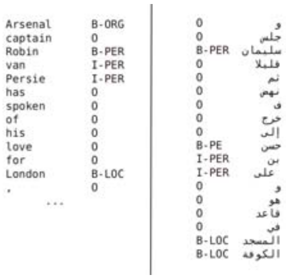
[1] The term Hadith is used to denote a saying or an act or tacit approval or criticism ascribed either validly or invalidly to the Islamic prophet Muhammad or other Islamic leaders. Hadith are regarded by traditional Islamic schools of jurisprudence as important tools for understanding the Quran and in matters of jurisprudence.

3. Preparing NoorCorp and NoorGazet

As reported in Conference on Computational Natural Language Learning in 2003 (Tjong Kim Sang and De Meulder, 2003), a corpora for named entity task must contains words with theirs NER tags. Same classes those was defined in Message Understanding Conference, contains organizations, locations and person names. Other types of named entities are tagged with MISC. Therefore, each word must be tagged with one of these tags:

- B-PERS: Beginning of a person name.
- I-PERS: Inside or ending of a person name.
- B-LOC: Beginning of a location name.
- I-LOC: Inside or ending of a location name.
- B-ORG: Beginning of a organization name.
- I-ORG: Inside or ending of a organization name.
- B-MISC: Beginning of a name which is not a person, location or organization name.
- I-MISC: Inside or ending of a name which is not a person, location or organization name.
- O: Other words.

In CoNLL, the decision has been made to keep a same format for training and test data for all languages. The format consists of two columns: the first one for words and second one for tags. Figure 1 shows two examples of standard corpora for NER task. In left side, a piece of tagged English text is shown. Often a TAB or SPACE character is used between word and its tag. Each word is written with its tag or tags in one separate line. All punctuation marks are written in separate line just like the other words. To make a proper corpora for NER task in Arabic religious texts, 3 corpora have been prepared from tagged text in Computer Research Center of Islamic Sciences which is located in Qom, Iran. These 3 corpora have been made from 3 books:



```
Arsenal    B-ORG
captain    O
Robin      B-PER
van        I-PER
Persie     I-PER
has        O
spoken     O
of         O
his        O
love       O
for        O
London     B-LOC
.          O
      ...
```

```
O       و
O       جلس
B-PER   سليمان
O       قليلا
O       ثم
O       نهض
O       ف
O       خرج
O       إلى
B-PE    حسن
I-PER   بن
I-PER   على
O       و
O       هو
O       قاعد
O       في
B-LOC   المسجد
B-LOC   الكوفة
```

Figure 1: Standard English and Arabic corpora for NER task

- A historical book, “Vaghat-a Seffeyn” written by “Nasr bin Mozahim Manghari” in 828 A.D.
- A traditional Hadith book, “El-Irshad fi marefati hojajellah alal-ibad” written by “Mohammad bin Mohammad Mofid” in 846 A.D.
- A jurisprudential book, “Sharaye el-Islam fi masaele harame val-halal” written by “Jafar bin Hasan” in 1292 A.D.

These corpora are compared based on number of words (after tokenizing), ratio of person, location and organization names in table 1. Also those are compared with ANERCorp (which contains newspaper data) in that table.

Table 1 shows these 4 corpora which belongs to modern and ancient Arabic text, are very different in their structures. In addition to differences between modern and ancient Arabic texts, ratio of using person and location names are different. In newspaper data, this ratio is high, but in compare to historical data, ratio of using names are lower than historical text. In traditional Hadith book, ratio of person names are higher, but names of locations are lower than mentioned texts. In jurisprudential texts, ratio of proper names are very low. In that corpora, the portion of proper names is lesser than 1 percent (totally 233 proper names in 48582 words). Ratio of each type of proper names are shown in table 2.

Corpus	Person	Location	Organization	Misc
Seffeyn	27.98%	43.52%	28.5%	0%
El-Irshad	80.66%	6.03%	13.03%	0%
Sharaye	30.19%	69.81%	0%	0%
ANERcorp	38.98%	30.42%	20.58%	10.01%

Table 2: Ratio of each types of proper names in NoorCorp and ANERCorp

Gazetteers are many important resources to improve the results of NER task. To make a perfect gazetteer, about 88000 proper names were gathered from “Jamiál-AHadith” software which has been produced by Computer Research Center of Islamic Sciences. Then we tokenized these names to their elements. For example “Hasan bin Ali bin Abdellah bin el-Moghayrah” was tokenized to 6 unrepeated elements: “Hasan”, “bin”, “Ali”, “Abd”, “Allah” and “Moghayrah”. These elements were produced for all proper names and added to a database with their frequencies. Finally a database with 18238 names were produced.

4. Noor ANER System

Noor ANER is a system based on conditional random fields which analyzes input text and extracts proper names after three types of preprocessing. We describe Noor ANER and its structure in this section.

4.1. Conditional Random Fields

Conditional random fields is a statistical modeling method which often is used in pattern recognition. Precisely, CRF is a discriminative undirected probabilistic graphical model.

4.1.1. Log-Linear Models

Let x as an example and y as a possible tag for that. A log-linear model supposes

$$p(y|x;w) = \frac{e^{\sum_j w_j F_j(x,y)}}{Z(x,w)} \tag{1}$$

Corpus	Number of words	Person	Location	Organization	Misc	Subject
Seffeyn	235842	6.47%	10.06%	6.59%	0%	History
El-Irshad	134316	14.31%	1.07%	2.36%	0%	Hadith
Sharaye	48582	0.48%	1.11%	0%	0%	jurisprudence
ANERcorp	150285	4.28%	3.34%	2.26%	1.10%	Newspaper

Table 1: NoorCorp and its containing books.

that Z is named as "partition function" and it equals with

$$Z(x,w) = \sum_{y'} e^{\sum_j w_j F_j(x,y')} \tag{2}$$

Therefore, having input x, predicted tag from model will be

$$\hat{y} = argmax_y p(y|x;w) = argmax_y \sum_j w_j F_j(x,y) \tag{3}$$

each of $F_j(x,y)$ are feature functions.
CRF model are a specific type of Log-Linear models. CRF in this article, refers to Linear-chain CRF.

4.1.2. Induction and training in CRF models

Training of CRF model means finding wight vector w such that make best possible prediction for each training example $\bar{x}$:

$$\bar{y}^* = argmax_{\bar{y}} p(\bar{y}|\bar{x};w) \tag{4}$$

However, before describing training phase, we must consider two main problems exists in induction phase: First, how can we compute 4 equation for each $\bar{x}$ and each set of weights w efficiently. This computation is exponential due to number of different sequences for tags $\bar{y}$. second, having $\bar{x}$ and $\bar{y}$ we must evaluate these values:

$$p(\bar{y}|\bar{x};w) = \frac{1}{Z(\bar{x},w)} e^{\sum_j w_j F_j(\bar{x},\bar{y})} \tag{5}$$

problem in here is denominator, because that needs all of sequences $\bar{y}$:

$$Z(\bar{x},w) = \sum_{\bar{y}'} e^{\sum_j w_j F_j(\bar{x},\bar{y}')} \tag{6}$$

for both of these problems, we needs efficient innovative methods, which without moment processing on each $\bar{y}$ in 6, processes all of them efficiently. The assumption that each feature function in this CRF models are dependent to two adjacent tags, aim us to resolve this problems. You can refer to (Elkan, 2008), (Lafferty et al., 2001) or (Sutton and McCallum, 2007) for more information.
When we have a set of training examples, we suppose our goal is finding parameters w_j so that conditional probability of occurring those training examples would be maximum. For this propose, we can use ascending gradient method. Therefore we need to compute conditional likelihood for a training example for each w_j. maximizing p is same as maximizing $ln\ p$:

$$\begin{aligned}\frac{\partial}{\partial_{w_j}} ln\ p(y|x;w) &= F_j(x,y) - \frac{\partial}{\partial_{w_j}} logZ(x,w) \\ &= F_j(x,y) - E_{y'\sim p(y'|x;w)}[F_j(x,y')].\end{aligned} \tag{7}$$

In other words, partially derivation to ith weight is value of ith feature function for true tag y minus average value of feature function for all of possible tags y'. Note that this derivation allows real value for each feature function, not only zero and one values. When we have all of training examples T, gradient ascending of condition likelihood, will be the sum of ascending for each training examples. Absolute maximum of all of these ascending are equal to zero, Therefore:

$$\sum_{\langle x,y\rangle \in T} F_j(x,y) = \sum_{\langle x,.\rangle \in T} E_{y\sim p(y'|x;w)}[F_j(x,y)] \tag{8}$$

This equation is correct for all of training examples not for each of them. Left side of above equation is total value of feature function j on all of training sets. Right side is total value of feature function j which is predicted by model. Finally, when we maximize conditional likelihood with online ascending method, adjustment of weight w_j would be calculated with this formula:

$$w_j := w_j + \alpha(F_j(x,y) - E_{y'\sim p(y'|x;w)}[F_j(x,y')]) \tag{9}$$

4.2. Preprocessing methods

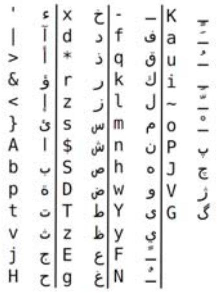
In training, testing and prediction phases we are using some preprocessing methods. In this section we describe about these methods.

4.2.1. Tokenizing

One of must useful preprocessing on text mining tasks, are tokenization. Tokenization is the process of breaking text up into words, phrases, symbols, or other **meaningful** elements called **tokens**. For example the word "Sayaktobounaha" in Arabic language, (which means "And they will write that") will be tokenized into "va+sa+ya+ktob+ooua+ha".
We have used AMIRA 2.1 software for tokenization process. We will describe more about that software in section 4.2.3..

4.2.2. Transliteration

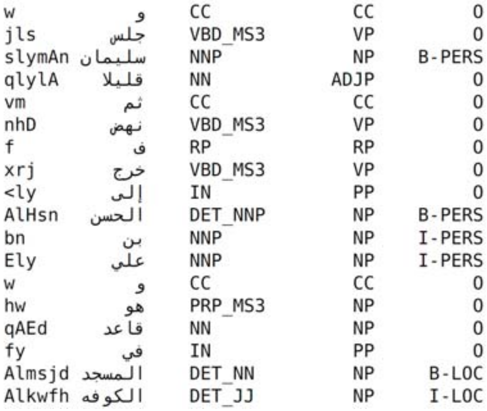
Another useful preprocessing method which often is last preprocess, is transliteration. Transliteration is replacing characters of first language with character of a destination language. Often second language is English. In this process, each character is mapped to one and just one character in destination language. In Noor ANER system we used Buckwalter transliteration. Figure 2 shows mentioned transliteration for Arabic and Persian languages (Habash et al., 2007). Figure 3 shows some examples for this transliteration. Second column from right, is real data in Arabic language and first column is transliterated data. Many general proposed language processing tools accept their inputs



'	ء	x	خ	-	ـ	K	ـٍ
|	آ	d	د	f	ف	a	ـَ
>	أ	*	ذ	q	ق	u	ـُ
&	ؤ	r	ر	k	ك	i	ـِ
<	إ	z	ز	l	ل	~	ـّ
}	ئ	s	س	m	م	o	ـْ
A	ا	$	ش	n	ن	P	پ
b	ب	S	ص	h	ه	J	چ
p	ة	D	ض	w	و	V	ڤ
t	ت	T	ط	Y	ى	G	گ
v	ث	Z	ظ	y	ي		
j	ج	E	ع	F	ـً		
H	ح	g	غ	N	ـٌ		

Figure 2: Buckwalter transliteration for Arabic language

in first column. For this reason the transliterated data is placed there.



```
w       و        CC          CC         O
jls     جلس      VBD_MS3     VP         O
slymAn  سليمان   NNP         NP    B-PERS
qlylA   قليلا    NN          ADJP       O
vm      ثم       CC          CC         O
nhD     نهض      VBD_MS3     VP         O
f       ف        RP          RP         O
xrj     خرج      VBD_MS3     VP         O
<ly     إلى      IN          PP         O
AlHsn   الحسن    DET_NNP     NP    B-PERS
bn      بن       NNP         NP    I-PERS
Ely     علي      NNP         NP    I-PERS
w       و        CC          CC         O
hw      هو       PRP_MS3     NP         O
qAEd    قاعد     NN          NP         O
fy      في       IN          PP         O
Almsjd  المسجد   DET_NN      NP     B-LOC
Alkwfh  الكوفه   DET_JJ      NP     I-LOC
```

Figure 3: Corpora after transliteration and adding POS and BPC tags

4.2.3. AMIRA software and part of speech tagging

AMIRA software has been developed by Mona Diab in Colombia University for standard Arabic language. AMIRA is a replacement for ASVMTools. This software contains a Tokenizer (TOK), a part of speech tagger (POS) and a base phrase chunker (BPC). The reports which were published in (Diab, 2009) shows this toolkit is very fast and reliable. Also user can adjust many different parameters in this software. AMIRA has been used in many papers about natural language processing in Arabic language. We have used this software toolkit in preprocessing phases of Noor ANER system.

4.2.4. Corpus preparation and training of CRF model

In Noor ANER system we have used FlexCRF, a general proposed implementation of conditional random fields. That software accepts the input in the following structure: The input must has three columns:

- First column, contains transliterated data. In our problem sequences for tagging process are sentences. Each sentence must ends with a period character. after each sentence, one line leaves blank.
- Second column consists feature functions. Structure of these feature functions has been described in documentation files of this software[2]. We are free to use any valid feature function sets for this column. But we must meet limitations of conditional random fields model. Therefore each feature function must depends on current word or predicate, up to two previous words or predicates and up to two next words or predicates. Our system uses these feature function templates:
  - One word.
  - Two consecutive words.
  - One predicate.
  - Two consecutive predicates.
  - Three consecutive predicates.
  - One word and one predicate.
  - Two predicates and one word.
  - Two words and one predicate.

  Predicates in Noor ANER are POS tags of each words. These POS tags are assigned by AMIRA software.
- Third column is NER tag for training and testing phases of CRF model.

As you can see in figure 3, we have different information for each word:

- Transliterated words (generated from original text).
- Original words (Typed by typists).
- POS tags (generated by AMIRA software from original text).
- BPC tags (generated by AMIRA software from original text).
- NER tags (verified by linguists)

The last column is needed for training and testing phases not in prediction phase.

4.3. Proper Name Candidate Injection

We described in previous section that predicates used in training of CRF model are POS tags. But indeed, predicates **are not exactly** POS tags. We have adjusted POS tags to improve the results in NER task. We enrich POS tags which are generated by AMIRA software from original input text:

1. If current word, is existed in our gazetteer, "NAME_" phrase is added to beginning of it POS tag. We named this word a "Proper Name Candidate".
2. If we encountered to two or more consecutive proper name candidates, we replace the POS tag with "NAME2" tag.

[2] Refer to this address for more information: http://flexcrfs.sourceforge.net/documents.html

In this approach, if total number of POS tags are n, the size of predicates will be $2n + 1$.

Why we expect better results with this approach? Importance of second item seems obvious. Many person names in Arabic languages also have adjective roles. But in major cases, when two or more of these word placed consecutively, we can tagged those as proper names, with very high probability. Especially, existence of relational words like "Bin" between them, raises this possibility. This probability was 94 percent in our experiments and based on this fact, we replace the POS tag to a new constant NAME2 tag here. First rule is very useful too. In fact we are producing a predicate that consists of POS information and a proposal to be a proper name. However, CRF model is deciding to tag this word as proper name or not. Using this approach, we generate extra predicates which have more probability to be proper names. But yet this is the CRF model which decides how to use this new informations.

With these descriptions, we expect to gain a reliable approach. Because when the POS tagger, AMIRA or any other software, tagged one word wrongly, the CRF models just ignores this tag sequence in training phase, because it don't find any more sentences with this wrong POS tag sequence. (Ignorance don't mean a complete ignorance here. CRF saves all of feature functions. but the possibility of using this new wrong tag sequence is very very low for a huge corpora) **Our experiences proved this claim**.

4.4. Structure of Noor ANER System

As we mentioned above, our tagged texts are converted to an standard transliterated NER corpora by some preprocessing tools. Then, we produced text with POS and NER tags using POS tagger software. Then another software generates predicates which enriched with proper name candidates. Generated resource after these processes is delivered to CRF trainer. Figure 4 shows this structure. In prediction phase, we have same process, but no NER tag is available in the resources.

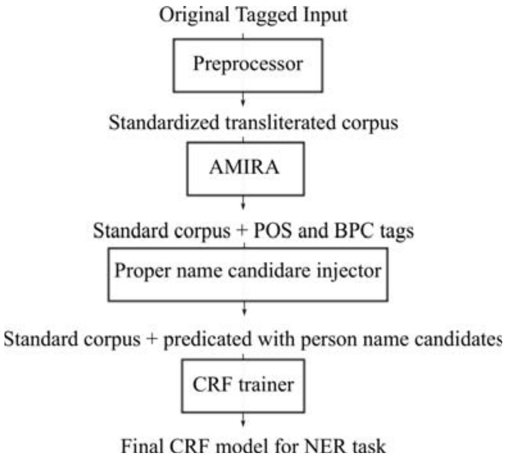

Original Tagged Input
Preprocessor
Standardized transliterated corpus
AMIRA
Standard corpus + POS and BPC tags
Proper name candidare injector
Standard corpus + predicated with person name candidates
CRF trainer
Final CRF model for NER task

Figure 4: Structure of Noor ANER System

Corpus	Topic	Precession	Recall	F-measure
Seffeyn	History	99.93%	99.93%	99.93
Al-Irshad	Hadith	95.62%	92.16%	93.86
Sharaye	Jurisprudence	100.00%	60.87%	75.68

Table 3: Evaluation of Noor ANER system on NoorCorp

5. Evaluation

We introduced 3 new corpora in section which are produced for Arabic NER task. Those corpora contains 3 different topics: history, Hadith and jurisprudence. Since Noor ANER has focused on person names, the results are shown just for person names. As table 3 shows, precession and recall metrics are very high for historical and traditional Hadith data. One of most reasons to obtain this high accuracy is existence of full names (that contains first name. father's name, ancestors name, nickname and even last name) in these topics. And full names consists their parts which are connected with some frequent relational words like "Bin". Therefore CRF model has a very strong pattern to extract many of person names.

Proper names in jurisprudence data are rare, thus extracting person names in this case is very very hard and not reliable. The results shows this fact.

6. Conclusion and future works

Results showed that Noor ANER act with very good performance on religious texts. The experiments declared we have very high F-measure for historical and Hadith data. Also we have produced 3 corpora based on three religious books in Arabic languages.

it is important to point out that we have used a language independent approach in development of our system. Although our system is based on a POS tagger like AMIRA, but the NER subtask in the system is language independent. Also There are many methods to generate POS tags with language independent approaches. Anyway, our method could adopt itself to any other languages which have an accurate POS tagger software.

Next generation of this system can be developed by using more feature functions and predicates which are created specially for Arabic language. Also we can add extracting other types of named entities to this system. For this cases, we need to make special gazetteers for names of locations and organizations.

As we mentioned in section 2., some of other systems for Arabic NER task, use hybrid models like combining multiple CRF models or even multiple methods to improve the results. Using such approaches can improve our system too.

7. References

H. Al-Jumaily, P. Martínez, J.L. Martínez-Fernández, and E. Van der Goot. 2011. A real time Named Entity Recognition system for Arabic text mining. *Language Resources and Evaluation*, pages 1–21.

Y. Benajiba and P. Rosso. 2007. ANERsys 2.0: Conquering the NER task for the Arabic language by combining the Maximum Entropy with POS-tag information.

In *Proc. of Workshop on Natural Language-Independent Engineering, IICAI-2007*.

Y. Benajiba and P. Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153.

Yassine Benajiba, Mona Diab, and Paolo Rosso. 2004. ARABIC NAMED ENTITY RECOGNITION: AN SVM APPROACH.

Y. Benajiba, P. Rosso, and J. BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. *Computational Linguistics and Intelligent Text Processing*, pages 143–153.

Mona Diab. 2009. Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. In Khalid Choukri and Bente Maegaard, editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April. The MEDAR Consortium.

Charles Elkan. 2008. Log-linear models and conditional random fields.

A. Elsebai and F. Meziane. 2011. Extracting person names from Arabic newspapers. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 87–89. IEEE.

H. Fehri, K. Haddar, and A.B. Hamadou. 2011. Recognition and Translation of Arabic Named Entities with NooJ Using a New Representation Model. In *International Workshop Finite State Methods and Natural Language Processing*, page 134.

Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter. 2007. On Arabic Transliteration. In Abdelhadi Soudi, Antal van den Bosch, Günter Neumann, and Nancy Ide, editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 15–22. Springer Netherlands.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Eighteenth International Conference on Machine Learning*.

Charles Sutton and Andrew McCallum. 2007. An Introduction to Conditional Random Fields for Relational Learning.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.