Hebrew Named Entity Recognition

Naama Ben Mordecai and Michael Elhadad
{benmorde,elhadad}@cs.bgu.ac.il
Department of Computer Science
Ben Gurion University of the Negev

Abstract

This paper presents a named entity recognition (NER) system for the Hebrew language. The Hebrew language has high morphological ambiguity, which makes automatic processing difficult. The first step in our work was to define the tagging task for the Hebrew language. Tagging guidelines were phrased and agreements tests were performed among human taggers. We present three models for approaching the NER problem in Hebrew: Hidden Markov Model, Maximum Entropy model and a simple model based on regular expressions and a lexicon extracted from the training corpus. Our main work was finding features suitable for the Hebrew language. The Maximum Entropy model has shown the best results out of the three. However, best results are achieved when combining the three models. The combined system has achieved good results, the best results achieved so far for Hebrew and comparable with those obtained for English.[1]

1 Introduction

The Named Entity Recognition (NER) task involves identifying noun phrases that are names, and assigning a class to each name. This task has great significance in the field of information extraction.
A considerable amount of work has been done in recent years on named entity taggers in many languages. The shared tasks of MUC-7, CoNLL-2002 and CoNLL-2003 concerned the NER problem. Recent work includes both knowledge engineering and machine learning approaches. Machine Learning approaches have the advantage of being dynamic (they adapt to new variety of text) and their development doesn't require expensive professional linguistic knowledge. Machine learning approaches include Maximum Entropy Models (ME), Hidden Markov Models (HMM) and support vector machines (SVM).
High accuracy results have been achieved in English. In English the problem is greatly simplified by the fact that most named entities start with a capital letter. In non-English languages reported performances is significantly lower.
In this paper, we investigate the NER task in Hebrew. The Hebrew language is characterized by high morphological ambiguity (close to 2 possible analyses per word on average), which makes automatic processing difficult. Hebrew names origin from different sources and use various transliteration methods. Many Hebrew names are ambiguous with nouns, verbs and adjectives. (For example, the name of the Israeli President is *katsav* which is ambiguous with the common noun meaning *butcher*). Hebrew doesn't have the advantage of using capital letters and the order of the words in a Hebrew sentence is quite free – making stochastic methods more difficult.
Another property of the Hebrew language is agglutination, a process which creates compound word made from other words. This raises the problem of segmenting a word to its original parts. This can not always be done with 100% accuracy.
Other aspects of the Hebrew language and culture influence the NER problem. For example: use of a Hebrew calendar and the transliteration of Latin acronyms and foreign names.

Our work presents three automatic models for Hebrew NER. Our main work was finding features that can cope with the unique character of the Hebrew language. Most of our features are automatically constructed and not based on professional knowledge. We focused on recognizing entity names (person, location organizations), temporal expression (date, time) and number expression (percent, money).

[1] This work is supported by the Knowledge Center for Hebrew Processing, Israel Science Ministry and the Frankel Fund.

There has not been a lot of work in the field of information extraction in Hebrew, so we had to establish some ground tools for approaching the NER task in Hebrew. We started with defining the NER task for Hebrew and writing guidelines suitable for Hebrew. Then we created a tagged corpus that was used for training and testing our system.

2 Experimental Data

We started our work with defining the NER task for Hebrew. We phrased special guidelines for Hebrew. The guidelines were adapted from the English guidelines (Chinchor, Brown, Ferro and Robinson, 1999) with adjustments and additions for the unique properties of Hebrew. We constructed agreement tests among human taggers. We refined the guidelines until a high level of agreement was reached. The guidelines are available in (Ben Mordecai 2005).

We used the guidelines to create a manually tagged a corpus. Our corpus consists of newspaper articles in different fields: news, economy, fashion and gossip. The articles where taken from a few known Hebrew newspapers. Our tagged corpus consists of about 57,000 words and 4,700 name expressions. It was split for training and testing our systems.

3 Baseline

The baseline system we created for Hebrew is based on set of regular expression patterns and a lexicon extracted from the training data. The regular expressions identify simple dates, time, percent and money expressions. The lexicon consists of named entities which appear in the training data. The system selects complete unambiguous names which appear in the lexicon.

Baseline results are shown in Table 1.

	Precision	Recall	F-measure
PER	59.03%	23.91%	34.03
LOC	93.22%	52.49%	67.16
ORG	61.22%	44.5%	51.54
DATE	46.66%	77.59%	58.28
TIME	77.27%	45.83%	57.54
MONEY	81.09%	83.93%	82.49
PERCENT	76.11%	91.67%	83.17
Overall	71.54%	48.46%	57.78

Table 1: baseline results

Results show that regular expression can effectively recognize money and percent expression. Date and time expression are recognized at some level, but there are many expressions that do not mach the regular expression.

We can see that some location expressions are unique and frequently used and that the preprocessed lexicon helps recognizing them.

Person names are the least recognized. As mentioned above, Hebrew names are various and can function as other parts of the speech. The Lexicon is not sufficient to recognize them in a sentence.

4 Knowledge Sources

The systems presented in this paper make use of the following knowledge sources:

Regular expressions: we defined this list to establish the baseline system.

Dictionary: we use a dictionary compiled from different internet sources. The dictionary consists of 7,000 named entity expression 1-3 words long. The different volumes of the dictionary are: first names, family names, dates (days, months, seasons, and holydays), countries, cities, locations, numbers, companies, organizations, money. In addition, the training data is automatically preprocessed to compile a number of lists:

Frequent Words List consists of words which appear in 5 or more articles in the training data.

Frequent Expression Lists are compiled for each name class and consists of expressions which appear in the training data 3 or more times.

Frequent Nouns Lists are compiled for each name class and consist of nouns which appear frequently before, after of within a name expression.

Part of Speech tagger and morphological disambiguator Our system uses a Part of Speech tagger and morphological disambiguator developed by Meni Adler (Ben Gurion University, 2005). Its error rate is about 8%.

5 Hidden Markov Model

5.1 Notation

We assume that a sequence of name classes can be modeled by a Markov process.

The HMM is denoted by a triplet $M = \{\Sigma, Q, \Theta\}$ where Σ is an alphabet of symbols; Q is a finite set of states, capable of emitting symbols from the alphabet Σ; Θ is a set of probabilities: state transition probabilities and emission probabilities. A path in the model is a sequence of states.

Given a sequence of symbols $w_1 \ldots w_n$ we can compute the most probable path $s_1 \ldots s_n$ that generated it.
We consider a simple HMM in which the state probabilities depend on the current symbol and the previous state. We assume that the transition and emission probabilities are independent:

$$P(w_i \mid s_i, s_{i-1}) = P(w_i \mid s_i) P(s_i \mid s_{i-1})$$

The probability of a path is:

$$P(s_1 \ldots s_n \mid w_1 \ldots w_n) = \prod_{i=1}^{n} P(w_i \mid s_i) P(s_i \mid s_{i-1})$$

In case we have training data in which we know the state sequences (tagged data), we can construct an HMM using maximum likelihood estimators. Once a model is constructed we calculate the most probable path for a given sequence using the Viterbi algorithm.

5.2 Hebrew NER using HMM

We experimented with several HMMs to approach the Hebrew NER problem. In each model we defined the alphabet and the state set differently. The following describes the HMM which produced the best results throughout our experiments.
States are defined as a product of the set of possible name classes and the set of possible Part of Speech tags. For example, there would be a state for PERSON + NOUN, PERSON + VERB etc. We also define special states for the beginning and end of a sentence. Overall we get a set of 212 states. The intuition for this state definition came from the fact that the syntactic structure of the sentence has great impact on the prediction of name classes. Defining the Part of Speech tags as part of the HMM states emphasizes the structure of the sentence through the transition probabilities.
As opposed to using the word itself as a symbol and defining the alphabet as all the words in the corpus, each state emits a string representing a product of several features of the word. This allows us to integrate more information in the model. The alphabet was defined by combining the following features for each word in the corpus: features which represent regular expressions; a feature which indicates whether the word appears within quotes; dictionary features for the current word and a window of ±2 words around it; features which indicate if the current word is in or a part of an expression in one of the pre-processed lists.
The HMM was constructed from the training data set. Transition and emission probabilities were calculated using maximum likelihood estimators. The Part of Speech tags were taken from the Part of Speech tagger.

5.3 HMM Experimental Results

Results of the described HMM NER system are shown in Table 2.

	Precision	Recall	F-measure
PER	82.41%	55.47%	66.31
LOC	86.81%	71.48%	78.4
ORG	73.4%	47.3%	57.53
DATE	85.58%	71.82%	78.1
TIME	24.42%	68.75%	36.04
MONEY	87.12%	66.96%	75.72
PERCENT	73.17%	63.59%	68.04
Overall	80.04%	59.41%	68.2

Table 2: HMM results

The error rate is reduced by 11% from the baseline results. The recognition of person names has improved by about 30%.
Defining the state set as a product of the name classes and the Part of Speech tags has helped the system recognize whole expressions. This definition enables the system to give the same name class to each word within one expression. Only 15% from the mistakes that were made were on part of an expression.
Experiments show that using a string of features produces better results than using the word itself as a symbol. For the prediction of name classes the information given by this set of features is much more important than the word itself. The disadvantage of this definition is that the features are considered as a union and not separately. The simple HMM only emits one symbol to each state.
A Limitation of this model is the use of the independence assumption which is not always correct in the NER problem.

6 Maximum Entropy Model

The Maximum Entropy (ME) probabilistic modeling technique has proved to be well adapted to cases where the model includes a large number of features. As apposed to the HMM, a ME model treats each feature separately. It gives each feature a weight according to its impact on the name class prediction.

6.1 The ME Approach

A ME system constructs a statistic model that is able to evaluate the likelihood of every word to be in one of several categories. The system estimates probabilities based on the principle of making as few assumptions as possible. Constraints are derived from training data expressing relationship between features and outcomes. We look for the probability distribution which is uniform except under the derived constraints. This is the distribution with the highest entropy out of all the distributions which satisfy our constraints.

The model computes the conditional probability $p(o \mid h)$ for any possible outcome *o* and history *h*. A history (or context) in a ME system is the data which link aspects of what we observe with a category we want to predict. The history is a set of binary features $f(h,o)$.
The distribution is unique and is built by a set of features and training data. It has the exponential form of:

$$p(o \mid h) = \frac{\prod_i \alpha_i^{f_i(h,o)}}{Z(h)}$$

$$Z(h) = \sum_o \prod_i \alpha_i^{f_i(h,o)}$$

The parameters α_i are estimated for every feature f_i from the training data by a procedure called Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972).
After creating a model, the beam search algorithm (Ratnaparkhi, 1998) is used to find the most probable sequence of outcomes.

6.2 Features

In the following section we present the features that were used by our ME system.
The current word is marked *w*, w_{-i} is the *i*th word before *w* and w_{+i} is the *i*th word after *w*.
Features with non-binary value are later transformed to binary features.
Structural Features: A set of binary features: *w* is the first word in the article; *w* is the first word in the sentence; *w* is the last word in the sentence.
Lexicon Features: *w* itself is used as a feature. This set contains one feature for each distinct word in the training data. Lexicon features are used for $w_{-2}, w_{-1}, w, w_{+1}, w_{+2}$. These features enable us to identify words which are frequently used inside a name expression, or particles which are used around such expressions.
Name Class of Previous Occurrence: Name class of *w*'s previous occurrence in the same article, as it has been computed by the model.
Name Class of Previous Words: Predicted name classes for w_{-2} and w_{-1}.
Dictionary Features: Dictionary features are taken for $w_{-2}, w_{-1}, w, w_{+1}, w_{+2}$ separately and for expression 2-3 words long around *w*. The value of each feature is the word/expression dictionary entry (that is, whether the word belongs to one of the dictionaries prepared during preprocessing).
Token Information: Features which represent information about the token *w* such as: 4 digit number, 2 digit number, contains digit, contains special character (@,$ etc.), contains percent sign, contains a dot, contains Latin character, can be read as a transliteration.
Regular Expression: The ME system uses the same regular expressions as the baseline system. Features indicate if *w* and the words around it match one of these regular expressions.
Word within quotes: Features which indicate sequence of tokens within quotes or brackets.

The following features make use of the Part of Speech tagger and morphological disambiguator
Part of Speech tags: Features which indicate the Part of Speech tags for $w_{-2}, w_{-1}, w, w_{+1}, w_{+2}$.
Lemma: Lemma is the dictionary entry of a word. It is the form of the word without its prefixes and suffixes. The lemma of *w* is used as a feature. This set contains one feature for each lemma in the training data.
Prefix and Suffix: The prefix and suffix of *w* are used as features.
Construct state: A binary feature which indicates if *w* is in construct state.
Frequent Words: A binary feature indicates if *w* is in the frequent words list.
Frequent Expression: If *w* is part of a frequent expression, a feature gets the value of its name class.
Frequent Nouns Lists Several features were defined to indicate whether *w* and the words around it are in one of the frequent nouns lists. The value of each feature is the list's name class.

6.3 ME Experimental Results

We studied the performance of our ME system with different feature combinations. Our results are presented in Table 4.

From all of our experiments, it appears that the most important features are the dictionary features and the Part of Speech features. Dictionary features are significant in recognizing organization and location names, Part of Speech features are significant in recognizing person names. We see that the lemma of a word does not provide enough information in predicting the name class of the word. Nevertheless, best results for the ME system are achieved by combining all the features mentioned above.

Final ME results are shown in Table 5. The error rate reduced by 20% from the baseline results. The precision rate of the system is high, but the recall is lower. This indicated that about 33% of the named entity expressions are not identified but those who do, are given the correct name class with high probability.

Feature ID	Feature description
A	POS tags
B	Dictionary features
C	Lemma
D	Preprocessing lists

Table 3: feature definition

Features used	Precision	Recall	F-measure
A	50.32%	22.68%	31.27
B	66.41%	31.18%	42.44
C	15.48%	0.13%	0.26
B+C+D	94.11%	48.43%	63.95
A+C+D	89.87%	59.7%	71.74
A+B+D	91.27%	63.91%	75.18
A+B+C	88.61%	63.65%	74.08

Table 4: overall performance with different features on the NE system

	Precision	Recall	F-measure
PER	91.6%	69.36%	78.94
LOC	92.13%	74.53%	82.4
ORG	82.96%	53.71%	65.2
DATE	92.04%	75.93%	83.21
TIME	47.14%	39.58%	43.03
MONEY	89.01%	80.36%	84.46
PERCENT	95.24%	81.67%	87.93
Overall	89.05%	67%	76.47

Table 5: ME results

Experiments with training data of different sizes show that using 90% of the training data will decrease the overall F-measure in 2%, and use of 60% of the training data will decrease the overall F-measure in 5%.

7 The Combined System

The ME system presents the best results out of the three. Still, recognition of organization and time expressions needs improvement. The recall rate of the ME system is still lower than the precision.

In order to improve our results we built a system which combines the three mentioned models. Each model has different qualities and exploits different knowledge sources indicating that a combined system can take advantage of them. In the baseline system the regular expressions dominate and the recognition of time, money and percent expressions is relatively high. The HMM succeeds in recognizing whole expressions. It recognizes different expressions than the ME model does. The experimental results of the ME are significantly higher. It succeeds in merging a large amount of features.

The best combination method would be a statistical one, a combination which assigns weight to each prediction according to a training data. Due to our limited resources, we could not set aside special training data for this purpose. The combination method we used is an empirical one. We use the method which produced the best experimental results.

Each system was trained on the same training data. Given a new text, the first stage in the process was sentence detection and tokenization. In the second stage, each system tagged the text separately. In the third stage we preformed a merge of the tagging results. The main principle of the merging method was to use the ME prediction and the other predictions as backup. Meaning, if the ME system didn't assign any name class to a word other predictions are taken into consideration. This principle had an exception in case of a prediction of the name class "time".

The combined system experimental results are presented in Table 6.

The error rate of the combined system is lower than 19% for most name classes (except organization). The combined system reduced the overall error rate of the ME system in about 2.5%, and the error rate of the baseline system in 21.3%. The recall rate has increased for all name

classes. As expected, the precision rate decreased as a result of the merging method.

	Precision	Recall	F-measure
PER	90.66%	73.82%	81.38
LOC	83.09%	82.8%	82.94
ORG	77.14%	62.03%	68.77
DATE	90.2%	85.18%	87.62
TIME	77.78%	87.5%	82.35
MONEY	85.71%	85.71%	85.71
PERCENT	97.83%	86.67%	91.91
Overall	84.54%	74.31%	79.1

Table 6: the combined system results

8 Conclusions

In conclusion, we have shown different approaches to the NER problem for the Hebrew language. We presented three different models: a simple baseline model based on a lexicon and regular expression, HMM and a ME model.
We have studied the impact of various features, specially suited for the Hebrew language, on the performance of our systems. We found that local features are the most important ones, our systems use features considering the current word and up to 2 words around it. The most dominant features out of our set of features are the dictionary features and Part of Speech tags. We need to consider the error rate of the Part of Speech tagger while integrating it in our system. Other features contribute to the name class prediction though they are not that dominate.
The ME model had shown the best results out of the three models presented. The best results were achieved by the combined system. These results are the best achieved so far for the Hebrew language.
The system is available on-line at http://www.cs.bgu.ac.il/~nlpproj.

References

Ben Mordecai, Naama: *Guidelines for Named Entity Tagging in Hebrew,* Technical Report, Dept of Computer Science, Ben Gurion University, 2005. Available at http://www.cs.bgu.ac.il/~nlpproj/NERtagging.pdf

Grishman, R: The NYU system for MUC-6 or where's the syntax?, In: *Proceedings of the Sixth Message Understanding Conference* (November 1995), Morgan Kaufmann.

D. Klein, J. Smarr, H. Nguyen and C. Manning: Named Entity Recognition with Character-Level Models. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003.

Robert Malouf: Markov models for language-independent named entity recognition. In: *Proceedings of CoNLL-2002*, Taipei, Taiwan, 2002, pp. 187-190.

A. Borthwick: A Maximum Entropy Approach to Named Entity Recognition. Ph.D. New York University. Department of Computer Science, Courant Institute, 1999.

R. Florian, A. Ittycheriah, H. Jing and T. Zhang: Named Entity Recognition through Classifier Combination. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003.

H. Chieu, H. Ng: Named Entity Recognition with a Maximum Entropy Approach. Department of Computer Science, National University of Singapore, 2003.

G. Lembersky: Named Entity Recognition in Hebrew & Hebrew Multiword Expressions: Approaches and Recognition Methods. Department of Computer Science, Ben Gurion University, 2003.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., and the annotation group: Algorithms that learn to extract information – BBN: Description of the SIFT system as used for MUC-7. *In proceedings of the Seventh Message Understanding Conference (MUC-7) (April 1998)*

Della Pietra, Stephen A., Vincent J. Della Pietra and Adam L. Berger "A Maximum Entropy Approach To Natural Language Processing", *IBM T. J.Watson Research Center*, 1996

Darroch. J., and Ratcliff, D., "Generalized iterative scaling for log-linear models." *The Annals of Mathematical Statistics 43* (1972) 1470-1480

Ristad, E.S. Maximum Entropy Modeling Toolkit (MEMT), release 1.6 beta, February 1998. Includes documentation, which has an overview of MaxEnt modeling.

Nancy Chinchor, Erica Brown, Lisa Ferro and Patty Robinson, *1999 Named Entity Recognition Task Definition*, MITRE, 1999.

C. Manning, H. Schutze: Foundations of Statistical Natural Language Processing. The MIT Press. Cambridge, MA. 1999.

Adler M., A Hebrew morphological disambiguator based on an unsupervised morpheme-based stochastic model, ISCOL 2005, Haifa.

Adwait Ratnaparkhi. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. *Ph.D. thesis*, University of Pennsylvania.