Named Entity Recognition in Turkish with Bayesian Learning and Hybrid Approaches

Sermet RehaYavuz, Dilek Küçük and Adnan Yazıcı

Abstract Named entity recognition is one of the significant textual information extraction tasks. In this paper, we present two approaches for named entity recognition on Turkish texts. The first is a Bayesian learning approach which is trained on a considerably limited training set. The second approach comprises two hybrid systems based on joint utilization of this Bayesian learning approach and a previously proposed rule-based named entity recognizer. All of the proposed three approaches achieve promising performance rates. This paper is significant as it reports the first use of the Bayesian approach for the task of named entity recognition on Turkish texts for which especially practical approaches are still insufficient.

1 Introduction

Named entity recognition (NER) is a well-established information extraction (IE) task and is defined as the extraction of the names of people, organizations, and locations, possibly with some temporal and monetary expressions [1]. Approaches to the NER task range from rule-based systems to learning-based and statistical systems which make use of annotated corpora and are therefore freed from human intervention as reviewed in [2]. Bayesian learning [3], hidden Markov models (HMMs) [4, 5], support vector machines (SVM) [6], and conditional random fields (CRF) [7] are

S. Reha.Yavuz · A. Yazıcı (✉)
Department of Computer Engineering, Middle East Technical University,
06800 Ankara, Turkey
e-mail: yazici@ceng.metu.edu.tr

D. Küçük
Electrical Power Technologies Group , TÜBİTAK Energy Institute,
06800 Ankara, Turkey
e-mail: dilek.kucuk@tubitak.gov.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_13,

among the widely employed machine learning/statistical techniques for several IE tasks including the NER task.

Considerable work on NER for well-studied languages such as English has been reported in the literature, yet, NER research on other languages including Turkish is quite rare. Among the related literature on Turkish texts, in [5], an HMM based statistical name extractor is described, in [8] the first rule-based NER system for Turkish is presented, the latter system is turned into a hybrid recognizer which is shown to outperform its rule-based predecessor [9]. A system utilizing CRF and a set of morphological features is presented in [10] and finally, a rule learning system for the NER task in Turkish texts is presented in [11].

In this paper, we target at the NER problem on Turkish texts and propose two approaches to address the problem: the former approach is based on Bayesian learning and the latter one is a hybrid approach combining the capabilities of the former approach with that of the rule-based named entity recognizer [8]. The evaluation results demonstrate that both of the presented approaches achieve promising performance rates on the evaluation corpora and the approaches are also compared with related literature.

The rest of the paper is organized as follows: in Sect. 2, the NER system employing the Bayesian learning approach is described and Sect. 3 presents the hybrid approach comprising two distinct hybrid systems with different characteristics. Evaluation results of the proposed approaches and their comparison with related work are provided in Sect. 4 and finally Sect. 5 concludes the paper.

2 The Bayesian Learning Approach for Named Entity Recognition in Turkish

The Bayesian learning approach proposed in this paper is a modified version of the Bayesian approach presented in [3]. It also utilizes the probabilities of tokens conforming to a set of features to be named entities, along with the probabilities of tokens used in the original Bayesian approach. In the following subsections, we first describe the original *BayesIDF* method proposed in [3] for information extraction and then present our approach based on this method.

2.1 BayesIDF Method

The Bayesian method, called *BayesIDF*, as described in [3] is based on the well-known Bayes' rule provided below. In classification problems, the denominator is usually ignored since it will be the same for all hypotheses and therefore, the rule simply states that the posterior probability of a hypothesis H is proportional to the product of the probability of observing the data conditioned on H, $Pr(D|H)$, and

the prior probability of H, $Pr(H)$.

$$Pr(H|D) = \frac{Pr(D|H)Pr(H)}{Pr(D)}$$

Within the context of information extraction, a hypothesis of the form $H_{p,k}$ corresponds to "k tokens beginning at position p to constitute a field instance" and out of all possible hypotheses the most probable one (with the highest $Pr(D|H_{p,k})Pr(H_{p,k})$) will be chosen [3]. In this Bayesian approach, $Pr(H_{p,k})$ is calculated as follows [3]:

$$Pr(H_{p,k}) = Pr(\text{position} = \text{p})\text{Pr}(\text{length} = \text{k})$$

In order to estimate the position, the instances in the training date are sorted based on their position, then grouped into bins of a certain size and frequencies for these bins are calculated, the position estimate for a test instance is found after interpolation between the midpoints of the closest bins. As the length estimate, the ratio of the number of instances of length k over all instances is used [3]. The $Pr(D|H_{p,k})$, the second probability necessary to calculate $Pr(D|H_{p,k})$ is found as follows where w is the number of tokens to be considered before and after an instance:

$$\left[\prod_{j=1}^{w} Pr(\text{before}_j = t_{p-j})\right]\left[\prod_{j=1}^{k} Pr(in_j = t_{p+j-1})\right]\left[\prod_{j=1}^{w} Pr(\text{after}_j = t_{p+k+j-1})\right]$$

In the training data set, for each token before/inside/after a field instance, the above probabilities are calculated as the ratio of the number the occurrences before/inside/after a field instance over all occurrences of a token [3].

2.2 Proposed Bayesian Approach

Our Bayesian approach for NER on Turkish texts uses the following modified formula to calculate $Pr(D|H_{p,k})$:

$$\frac{\sum_{j=1}^{BSD} Pr(before_j = t_{p-j}) + \frac{\sum_{j=1}^{k} Pr(in_j = t_{p+j-1})}{k_{avg}} + \sum_{j=1}^{ASD} Pr(after_j = t_{p+k+j-1}) + \frac{Pr(FC)}{k_{avg}}}{BSD + ASD + FC + 1}$$

In the above formula, $Pr(FC)$ is calculated as follows:

$$Pr(FC) = \sum_{f=1}^{FC}\sum_{j=1}^{k} \sigma(t_{p+j-1}, f)\phi(f)$$

where $\sigma(t_{p+j-1}, f)$ is 1 if t_{p+j-1} conforms to feature$_f$ and 0 otherwise and $\phi(f)$ is the probability that a token conforming to feature$_f$ is a named entity.

In the formula for $Pr(D|H_{p,k})$, *BSD* stands for *Before Surroundings Distance* and *ASD* stands for *After Surroundings Distance* which correspond to the number of tokens to be considered before and after named entity instances in the training set, respectively. As a matter of fact, we have used these parameters instead of the context parameter, w, in the original *BayesIDF* method [3] summarized in the previous subsection so that these parameters can independently be set to different values. We have also included in the formula the probabilities calculated for each of the enabled features as $Pr(FC)$, where the number of features is denoted as FC (for *feature count*). Features can also be assigned weights so that the corresponding probabilities are multiplied with certain coefficients corresponding to these weights. Instead of multiplying the probabilities (as the contribution of the occurrence of each of the tokens to the overall probability is assumed to be independent [3]), we have added them together and normalized the resulting summations by dividing them to $BSD + ASD + FC + 1$. The reason for using summations instead of products is that due to the scarcity of the available annotated corpora used for training, using products (as in [3]) has resulted in very low probabilities which in turn has lead to low success rates. We should also note that the probability summation regarding the inside tokens and $Pr(FC)$ are divided by the average number of tokens in a named entity, k_{avg}, as calculated from the training data set.

Below, we first describe the details of the parameters used during the training phase including *BSD*, *ASD*, and *case sensitivity*. Next, we describe the features (or feature sets when applicable) employed with pointers to related studies. It should again be noted that assigning different coefficients to distinct features we can alleviate or boost the effects of these features.

The parameters utilized by the modified Bayesian method proposed:

- *Before Surroundings Distance* (BSD): In any Bayesian technique, a number of tokens before and after a field instance are used for training and estimation which are called *surroundings*. *BSD* is the number of tokens before a named entity where probabilities for these tokens will be calculated and utilized during the calculation of the probability for a candidate named entity. To illustrate, when *BSD* is three, corresponding probabilities of three tokens before a named entity will be calculated during training phase and will be utilized during the estimation phase.
- *After Surroundings Distance* (ASD): As its name implies, *ASD* parameter is similar to *BSD* and it specifies the number of following tokens that will contribute to the probability of a candidate named entity to be classified so.
- *Case Sensitivity*: This parameter specifies whether each token should be considered in a case-sensitive or in a case-insensitive manner. For instance, if case-sensitivity is turned off, the tokens *bugün* and *Bugün* (meaning '*today*') will be considered as the same token during the calculation of the probabilities.

The features utilized by the proposed method:

- *Case Feature*: This feature is used to map each token to one of the four classes: *all-lower-case*, *all-upper-case*, *first-letter-upper-case*, and *inapplicable*. The last class is for representing those tokens comprising punctuation marks and/or numbers. This feature is especially important for candidate named entities as case

information is known to be a plausible clue for person, location, and organization names in several languages, including Turkish.

- *Length Feature*: Similar to the previous feature, this feature maps each token to different length-related classes: *zero-length*, *one-char*, *two-chars*, *three-chars*, *four-chars*, and *longer*. Again, the probability of being in a named entity for each of these classes is calculated and utilized during the training and estimation phases.
- *Alphanumeric Feature*: This feature maps each token to one of four classes according to the nature of characters included in the tokens and again the probability of being in a named is calculated for each of these classes, to be used during the estimation phase. These classes are: *all-alpha*, *all-numeric*, *alphanumeric*, and *inapplicable*.
- *NF (Nymble Features)*: This set of features comprises a subset of the features utilized by the Nymble NER system for English [4]. Nymble is a statistical system which uses a variant of the HMM and achieves successful results over the related literature [4]. This feature set, obtained from the feature set of the Nymble system, encompasses some of the features defined above as it includes the following: *two-digit-number*, *four-digit-number*, *alphanumeric*, *other-number*, *all-capital*, *first-capital*, *lower-case*, *other*.
- *Lexical Resource Feature*: The feature of appearance in a lexical resource (name lists, gazetteers, etc.) is also considered as a distinct feature, i.e., *lexical resource feature*. As the required resources; person, location, and organization name lists of the rule-based recognizer for Turkish [8] are utilized.

3 The Hybrid Approaches

We have also proposed two different hybrid named entity recognition systems which utilize the Bayesian learning based recognizer described in the previous section and the rule-based named entity recognizer [8]. These two hybrid systems are briefly described below:

1. *Training-Phase Hybrid System*: This hybrid system is first trained on all the available training data, as the Bayesian learning based recognizer. But before carrying out the estimations on the test data, the rule-based recognizer [8] is run on the test data and the resulting annotated test data is also utilized as additional training data to update the probabilities to be employed by the hybrid system.
2. *Estimation-Phase Hybrid System*: This hybrid system version utilizes the output of the rule-based recognizer [8] during the estimation phase as a distinct feature. The output of the rule-based recognizer is parsed and the tokens corresponding to the annotated named entities are assigned additional scores to be used along with the probabilities calculated during the training phase. When this hybrid system makes estimations on the test data, if all of the elements of a considered token group are annotated as named entities by the rule-based system, this token group

gets an additional score of 1.0 while a token group partially annotated by the rule-based system gets an additional score of 0.5.

4 Evaluation and Discussion

During the testing of the proposed system in Freitag's work [3], a threshold value is used so that tokens with posterior probabilities above this threshold value are annotated with the corresponding named entity types and the remaining tokens are not annotated. In our study, we have tried several different threshold values during testing and the best results obtained are reported in this section.

The performance evaluation of the proposed approaches has been carried out with the widely used metrics of precision, recall, and F-Measure. These metrics, as utilized in studies such as [9, 12, 13], also give credit to partial named entity extractions where the type of the named entity is correct but its span is not correct. The exact formulae for the metrics are presented below:

$$\text{Precision} = \frac{\text{Correct} + 0.5 * \text{Partial}}{\text{Correct} + \text{Spurious} + 0.5 * \text{Partial}}$$

$$\text{Recall} = \frac{\text{Correct} + 0.5 * \text{Partial}}{\text{Correct} + \text{Missing} + 0.5 * \text{Partial}}$$

$$\text{F}-\text{Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

In these formulae, *Correct* is the number of correctly-estimated named entities in terms of their type and span (i.e., their location and the number of tokens included). *Spurious* is the number of incorrectly-estimated named entities, that is, those estimated ones which are not in the answer key. *Missing* is the number of named entities which are missed by the recognizer though they exist in the answer key. Lastly, *Partial* denotes the number of partially-estimated named entities which are of the correct type but their spans are not correct [9, 12].

In order to train the Bayesian learning based recognizer and test its performance, we have used one of the data sets which was previously compiled and annotated with named entities [9]. This news text data set comprises 50 news articles from METU Turkish corpus [14] with a total word count of 101,700 where each article contains about 2,000 words. The annotated entities in this set encompass 3,280 person, 2,470 location, 3,124 organization names along with 1,413 date/time and 919 money/percent expressions, hence amounting to 11,206 named entities.

We have evaluated the performance of the proposed Bayesian learning based recognizer on this data set with ten-fold cross validation. During this evaluation, *BSD* and *ASD* are taken as 1 and among the feature set, only NF features are enabled. The ten-fold cross validation results of the recognizer are presented in Table 1.

Table 1 Ten-fold cross validation results of the Bayesian learning based named entity recognizer on the data set

Named Entity Type	Precision (%)	Recall (%)	F-Measure (%)
Person	94.41	86.54	90.30
Location	92.10	82.54	87.06
Organization	88.78	87.29	88.03
Date	90.07	77.19	83.13
Time	86.13	75.17	80.28
Money	76.65	94.71	84.73
Percent	88.27	96.09	92.01
Overall	*90.74*	*85.40*	*87.99*

Table 2 Ten-fold cross validation results of the Bayesian learning based named entity recognizer on the data set (for different recognizer configurations)

Configuration	Precision (%)	Recall (%)	F-Measure (%)	Effect
Baseline	74.67	74.32	74.49	
Baseline+Alphanumeric feature	74.42	74.20	74.31	Negative
Baseline+Case feature	75.08	74.53	74.80	Positive
Baseline+Case sensitivity	75.31	75.85	75.58	Positive
Baseline+Length feature	74.31	73.89	74.10	Negative
Baseline+Lexical resource feature	70.75	75.77	73.17	Negative
Baseline+NF	74.95	74.32	74.63	Positive

In order to test the individual contribution of each the parameters/features, we have carried out evaluations, first on a baseline configuration of the recognizer, and then turning each of the parameters/features on, while turning the remaining ones off. During the evaluations, the coefficients are all set to 1.0 and in the baseline configuration, *BSD* and *ASD* are 0 and all of the features are turned off. The corresponding evaluation results are provided in Table 2 where the first row corresponds to the performance results of the recognizer in the baseline configuration. The last column denotes the effect of turning the corresponding parameter/feature on, to the overall F-Measure.

Experiments with different *BSD* and *ASD* as integers within the [1–5] scale have shown that the best F-Measure rates are obtained when *BSD* and *ASD* are both 1. Increasing these values have positive effects for some named entity types while having negative effects on others, and the overall F-Measure rates corresponding to higher *BSD* and *ASD* values are less than those rates when *BSD* and *ASD* are both set to 1.

We have used another news article from METU Turkish corpus [14] as the test set to evaluate and compare our Bayesian learning based recognizer as well as the two hybrid recognizers described in Sect. 3 built on the top of the former system, with the previously proposed rule-based system [8] and its hybrid counterpart [9]. The systems that require training have been trained on the aforementioned news text

Table 3 Evaluation results of the proposed named entity recognizers and related work.

Named entity recognizer	Precision (%)	Recall (%)	F-Measure (%)
Rule-based recognizer [8]	94.71	81.36	87.53
Hybrid (Rule-based+Rote learning recognizer [9])	94.27	81.9	87.65
Bayesian learning based recognizer	96.16	81.82	88.41
Training phase hybrid recognizer	92.68	87.56	90.05
Estimation phase hybrid recognizer	93.38	89.57	91.44

data set (with a word count of 101,700). The news article used as the new test data set has 2057 words and after its annotation to create the answer key, it comprises 228 annotated named entities where there are 102 person, 42 location, and 71 organization names along with 11 date expressions, 1 monetary and 1 percent expressions with no instance of time expressions. The evaluation results of the newly proposed two recognizers and those recognizers previously proposed for Turkish texts are presented in Table 3. The performance evaluations of the named entity recognizers proposed in the current paper are given in the last three rows.

The results in Table 3 show that the proposed Bayesian learning based recognizer achieves better results than the rule-based and hybrid (rule-based+rote learning) systems previously proposed [9]. The hybrid systems proposed, in turn, achieve better performance rates than their predecessor, the Bayesian learning based recognizer. Among the proposed hybrid systems, the latter estimation-phase hybrid system achieves higher success rates than the training-phase hybrid system. However, as pointed out at the beginning of this section, it should be noted that the proposed three approaches do not used predefined threshold values, instead, they try several alternative values during testing and the highest results achieved by the systems are given in Table 3. Therefore, in order to make a more appropriate comparison with the related work, a predetermined threshold value should be obtained (either heuristically or through a learning procedure) and utilized during the testing of the proposed systems, as future work.

To summarize, all of the proposed systems achieve promising results on the test data set which is a significant contribution to NER research on Turkish texts, as related research is quite insufficient compared to studies on languages such as English and, to the best of our knowledge, the proposed systems are the first to apply a Bayesian approach to this task on Turkish texts with a limited training data set. Yet, we expect that the results should be verified on larger test corpora and can be improved by increasing the annotated training data set, both of which are plausible future research directions. Other important future research topics include deeper elaboration of the employed parameters and features on larger corpora to better evaluate their effects.

5 Conclusion

Named entity recognition, as the other information extraction tasks, gains more significance every day, mostly due to the increase in the size of natural language texts, such as those on the Web, that need to be processed. In this paper, we target at named entity recognition in Turkish texts and propose two approaches for this problem: the first one is a Bayesian learning based approach and the second approach comprises two hybrid recognizers with different characteristics where the Bayesian learning system is basically utilized together with a previously proposed rule-based recognizer to achieve better performance rates. The evaluation results have shown that the proposed three approaches achieve promising results and the two hybrid approaches perform better than the Bayesian learning based recognizer. Yet, in order to further verify and increase the success rates of the proposed approaches, larger annotated corpora are necessary and the lack of such corpora is known to be one of the main problems against information extraction research on Turkish texts.

References

1. Grishman R (2003) Information extraction. In: Mitkov R (ed) The Oxford Handbook of Computational Linguistics. Oxford University Press, Oxford
2. Turmo J, Ageno A, Catala N (2006) Adaptive information extraction. ACM Comput Surv 38(2):1–47
3. Freitag D (2000) Machine learning for information extraction in informal domains. Mach Learn 39(2–3):169–202
4. Bikel DM, Miller S, Schwartz R, Weischedel R (1997) Nymble: a high-performance learning name-finder. Proceedings of the Fifth Conference on Applied Natural Language Processing, In, pp 194–201
5. Tür G, Hakkani-Tür D, Oflazer K (2003) A statistical information extraction system for Turkish. Nat Lang Eng 9(2):181–210
6. Li Y, Bontcheva K, Cunningham H (2009) Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. Nat Lang Eng 15(2):241–271
7. McCallum A, Li W (2003) Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Proceedings of the Seventh Conference on Natural, Language Learning, In, pp 188–191
8. Küçük D, Yazıcı A (2009) Named entity recognition experiments on Turkish texts. Proceedings of the International Conference on Flexible Query Answering Systems, In, pp 524–535
9. Küçük D, Yazıcı A (2012) A hybrid named entity recognizer for Turkish. Expert Syst Appl 39(3):2733–2742
10. Yeniterzi R (2011) Exploiting morphology in Turkish named entity recognition system. Proceedings of the ACL Student Session, In, pp 105–110
11. Tatar S, Çicekli İ (2011) Automatic rule learning exploiting morphological features for named entity recognition in Turkish. J Inf Sci 37(2):137–151

12. Küçük D, Yazıcı A (2011) Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos. Knowl Based Syst 24(6):844–857
13. Maynard D, Tablan V, Ursu C, Cunningham H, Wilks Y (2001) Named entity recognition from diverse text types. In: Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)
14. Say B, Zeyrek D, Oflazer K, Özge U (2002) Development of a corpus and a treebank for present-day written Turkish. In: Proceedings of the 11th International Conference of Turkish, Linguistics (ICTL)