

Name Tagging Using Lexical, Contextual, and Morphological Information

Gökhan Tür, Dilek Z. Hakkani-Tür, Kemal Oflazer

Department of Computer Engineering
Bilkent University, Bilkent, Ankara, TR-06533, Turkey
{tur,hakkani,ko}@cs.bilkent.edu.tr

Abstract

This paper presents a probabilistic model for automatically tagging names in a Turkish text. We used four different information sources to model names, and successfully combined them. Our first information source is based on the surface forms of the words. Then we combined the contextual cues with the lexical model, and obtained a significant improvement. After this, we modeled the morphological analyses of the words, and finally, we modeled the tag sequence, and reached an F-measure of 91.56% in Turkish name tagging. Our results are important in the sense that, using linguistic information, i.e. morphological analyses of the words, and a corpus large enough to train a statistical model helps this basic information extraction task.

1. Introduction

A basic task of information extraction (IE) from text or speech is marking names (persons, locations, and organizations), and certain structured expressions (monetary values, percentages, dates and times). This is known as *named entity (NE) extraction* task. In this task, finding only the names is called *name tagging*.

Named entity extraction task has been introduced by DARPA, and evaluated as an understanding task in both the Sixth and Seventh Message Understanding Conferences (MUC-6 (1995) and MUC-7 (1998)). A very detailed definition of the named entity extraction task has been developed in the framework of these programs (Chinchor and Robinson, 1998).

Name tagging task categorizes the names by their types as follows:

- **ORGANIZATION:** named corporate, governmental, or other organizational entity,
- **PERSON:** named person or family,
- **LOCATION:** name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)

IE systems are usually evaluated and compared using broadcast news transcriptions, where there are lots of named entities. Consider Figure 1 for an example news piece where the names are marked by SGML tags.

Although name tagging seems like a very straightforward process, even human annotators have a performance of 98%-99%. In real text, there are lots of cases in which it is very hard to determine the type of the name (for example determining whether *Washington* is a location or a person), or even whether it is a name or not (for example *Dow Jones* is not a name). This is why the official guideline of this task is very detailed, and tries to capture all kinds of such cases.

For this task, a corresponding Turkish example is provided in Figure 2.

Note that, the morphemes added after the names are not considered to be a part of the name, in order to be consistent with its definition for English. In English there are only a few such cases, (such as "Fred's"), but since Turkish is a

highly agglutinative language, theoretically there are infinite number of word formations. In fact, we separated the suffixes after the apostrophe and handled them as a separate word while training our models. This enabled us to prevent data sparseness due to the agglutinative nature of the language, and gave cues about the type of the name.

2. Previous Work

Similar to most other language processing systems, developers have approached the named entity extraction problem as one of building a hand-crafted rule-based system, an automatically trained system, or a combination of these two approaches.

2.1. Rule-based Approaches

Perhaps the most famous rule-based information extraction system is FASTUS, a slightly permuted acronym for Finite State Automaton Text Understanding System, developed by SRI International, Artificial Intelligence Center (Hobbs et al., 1996). FASTUS is a set of cascaded non-deterministic finite-state automata, hence it is very fast. FASTUS was one of the best performing systems in MUC-6, with an F-measure of 94% in the named entity extraction task.

The first commercial product, emerged from the named entity extraction task is called NetOwl Extractor from IsoQuest formed by SRA International (Krupka and Hausman, 1998). This system has rules consisting of a pattern and an action. Although this system outperformed all other systems participating in MUC-6 with an F-measure of 96.42%, its performance was 91.32% in MUC-7.

2.2. Machine Learning Approaches

The most successful statistical system participating MUC-7 for this task was BBN's IdentiFinder System (Bikel et al., 1999; Miller et al., 1998), with an F-measure of 90.61%. IdentiFinder uses a hidden Markov model (HMM), that learns to recognize and classify name classes (names, dates, times, and numerical quantities, etc.). The conceptual structure of this HMM is depicted in Figure 3.

New York University (NYU) participated in MUC-7 with a new system called MENE (Maximum Entropy Named Entity) (Borthwick et al., 1998a; Borthwick et al.,

... Good evening from <ENAMEX TYPE="LOCATION"> Havana </ENAMEX> where one of the day's big stories has begun to unfold. One of them the Pope is here and the world is waiting to see whether he will shake up this island and the veteran communist leader who runs it <ENAMEX TYPE="PERSON"> Fidel Castro </ENAMEX>. The other very big story of the day is in <ENAMEX TYPE="LOCATION"> Washington </ENAMEX> where the <ENAMEX TYPE="ORGANIZATION"> White House </ENAMEX> administration has already been badly shaken up by the possibility that president <ENAMEX TYPE="PERSON"> Clinton </ENAMEX> and one of his advisors <ENAMEX TYPE="PERSON"> Vernon Jordan </ENAMEX> obstructed justice. ...

Figure 1: An example broadcast news word transcript, with named entities are marked.

... 3 yılda bir yapılan <ENAMEX TYPE="ORGANIZATION"> Dünya Enerji Konseyi </ENAMEX> Kongresi'nde en seçkin beyinler , enerji sorununun yarımını tartışacak. Dün bir kokteyl ile başlayan kongrede bugün <ENAMEX TYPE="LOCATION"> ABD </ENAMEX> Başkanı <ENAMEX TYPE="PERSON"> Clinton </ENAMEX> , <ENAMEX TYPE="LOCATION"> Arjantin </ENAMEX> Devlet Başkanı <ENAMEX TYPE="PERSON"> Carlos Menem </ENAMEX> ve <ENAMEX TYPE="LOCATION"> Güney Afrika </ENAMEX> Başkan Yardımcısı <ENAMEX TYPE="PERSON"> Thabo Mbeki </ENAMEX> konuşacak...

Figure 2: An example Turkish news article, whose named entities are marked.

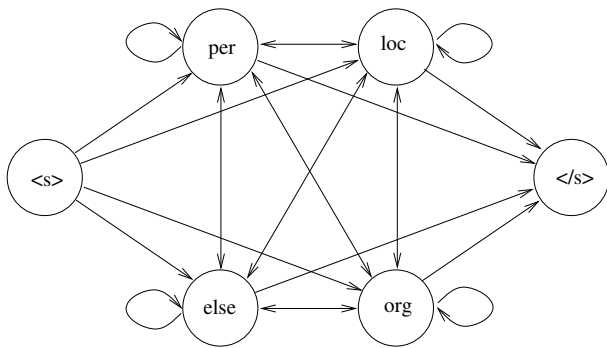


Figure 3: The conceptual structure of the basic HMM used by BBN for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, per denotes person, loc denotes location, org denotes organization, and else denotes that it does not belong to any of these categories.

1998b), employing maximum entropy models. MENE reached an F-measure of 89% in MUC-7 official evaluations.

MITRE participated MUC-6 evaluations with the Alembic system (Aberdeen et al., 1995), using transformation-based error-driven learning algorithm of Brill (1993). Their performance was in mid 80s.

Bennett et al. (1997) used binary decision trees using C4.5 (Quinlan, 1986) for the name tagging task. The decision tree decided whether it is a name boundary or not. They used features indicating semantic properties, locations, part-of-speech tags, and token types. Their performance was 88.1% in the MUC-6 evaluation set.

An independent study by Cucerzan and Yarowsky (1999) attempts to build a language independent name tagger using a bootstrapping algorithm based on iterative learning. Re-estimation of contextual (e.g. "Mr.", "mayor of") and word-internal (e.g. "-oğlu" is a typical surname indicator in Turkish) patterns are captured in hierarchically smoothed trie models. This algorithm was evaluated for

Romanian, English, Greek, Turkish, and Hindi. For Romanian, using a training set of 12,320 words, they reached an F-measure of 70.47%. For Turkish, training set was 5,207 words, and the final F-measure was 53.04%. This work was important in the sense that it was the first attempt for name tagging of Turkish. According to us, this algorithm is a novel approach, but needs more training data.

2.3. Hybrid Approaches

Among all other systems, the best performing system with an F-measure of above 94% at MUC-7, has employed a hybrid approach. The HCRC Language Technology Group (LTG) from the University of Edinburgh had an outstanding performance in recent MUC-7 evaluations (Mikheev et al., 1998). LTG worked in 5 phases, where first and third phases were rule-based, other phases relied on a pre-trained maximum entropy model.

Another hybrid system was developed by University of Manitoba of Canada for MUC-7 (Lin, 1998). They augmented the manually coded pattern rules with the rules extracted from the training data using a collocation database. Then they used contextual cues to tag the unknown words using a Naive-Bayes classifier. They reached an F-measure of 86.07% in MUC-7.

3. Approach

Our approach is based on n -gram language models embedded in hidden Markov models. We used the following 4 models in the name tagging task:

- *Lexical Model*, which captures the lexical information using only word tokens.
- *Contextual Model*, which captures the contextual information using the surrounding context of the word tokens. This model is especially helpful in tagging unknown words.
- *Morphological Model*, which captures the morphological information with respect to the corresponding case

and name tag information. In order to build this model, we used the morphological parses of the words.

- *Name Tag Model*, which captures the name tag information (person, location, organization, and else) of the word tokens.

Each model is smoothed using the Good-Turing method (Good, 1953) combined with the back-off modeling proposed by Katz (1997). In this work, in order to build a language model, and decode the most probable output in an HMM with the Viterbi algorithm (Viterbi, 1967), we used the publicly available SRILM toolkit, developed by Andreas Stolcke (Stolcke, 1999).

We would like to explain each model in detail in the following subsections, then discuss on the methods for combining these 4 models.

3.1. Lexical Model

For lexical modeling, we used a simplified version of BBN’s name finder (Bikel et al., 1999). The states of the hidden Markov model were word/tag combinations, where the tag indicated whether a word was part of a proper name, and of what type (person, place, or organization). Transition probabilities consisted of trigram probabilities over these combined tokens. The word/tag observation likelihoods for each state was set to 1.

In order to detect the boundaries of the names, we used a fictitious boundary flag. This flag holds one of the following 3 values:

1. *yes*: indicates that there is a name boundary.
2. *no*: indicates that there is no name boundary.
3. *mid*: indicates that the previous and the next tokens belong to the same name.

The conceptual structure of this HMM is depicted in Figure 4. Note that, although it is possible to get a sequence of “person mid organization”, the use of language model discourages such transitions for all cases. This is why we did not need to put a separate “mid” boundary state for each of these 3 name types.

An example will clarify this notation. Consider following piece of annotated text:

```
<ENAMEX TYPE="ORGANIZATION"> Bilkent
University</ENAMEX>'s <ENAMEX TYPE=
"ORGANIZATION">Graduate School of Business
</ENAMEX> is in Ankara.
```

The corresponding output sequence for this text would be as follows:

```
"<s> boundary/yes Bilkent/organization boundary/mid
University/organization boundary/yes 's/else boundary/yes
Graduate/organization boundary/mid School/organization
boundary/mid of/organization boundary/mid Busi-
ness/organization boundary/yes is/else boundary/no in
boundary/yes Ankara/location boundary/yes </s>"
```

This means, name tagging task does not only require tagging each word with one of the 4 possible tags (person,

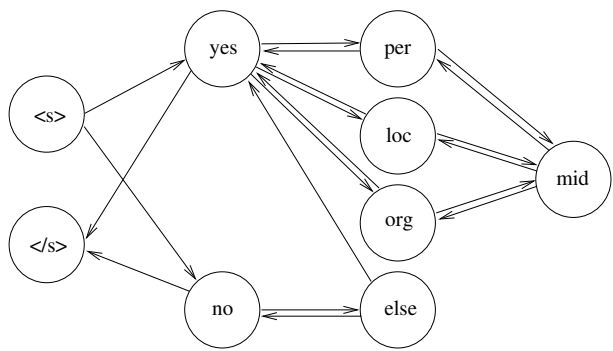


Figure 4: The conceptual structure of the basic HMM for name tagging. *<s>* denotes the start of sentence, and *</s>* denotes the end of sentence, *yes* denotes the name boundary, *no* denotes that there is no name boundary, *mid* denotes that it is in the middle of a name, *per* denotes person, *loc* denotes location, *org* denotes organization, and *else* denotes that it does not belong to any of these categories.

location, organization, and else), but also requires detecting the boundaries.

In fact, using this boundary flag improved the tagging performance. This flag has also performed as a connection between the surrounding tokens. Consider the following example:

```
<ENAMEX TYPE="ORGANIZATION">Ankara
Üniversitesi</ENAMEX>
```

The city “Ankara” can either be location or a part of an organization. As seen from the Table 1, the boundary flag helps us to find the correct tagging, for example the trigram “Ankara/organization boundary/mid Üniversitesi/organization” is about 4000 times more probable than the trigram “Ankara/location boundary/yes Üniversitesi/organization”, although tagging “Ankara” as location is more probable. The reason for this difference is that there is no occurrence of the bigram “boundary/yes Üniversitesi/organization”, but lots of “boundary/mid Üniversitesi/organization”.

3.2. Contextual Model

For contextual modeling, we improved our lexical language model as follows: We marked as unknown every other word in our training data, and then built a language model, then interpolated this model with the lexical model. Using this contextual model, we could tag the unknown words by looking at the context. This idea has first been used in (Hakkani-Tür et al., 1999), and explained in detail in (Tür, 2000 forthcoming). For example, the word after the abbreviation “Dr.” is generally a person, The word “University” is often a part of an organization.

In order to demonstrate this model with a real example, consider this piece of text:

```
Dr. <ENAMEX TYPE="PERSON"> Tür </ENAMEX>
```

Assuming that the word “Tür” is unknown, i.e. did not appear in the training data, we can use the contextual model

Output Sequence	Probability
Ankara/ <i>organization</i> boundary/ <i>mid</i> Üniversitesi/ <i>organization</i>	0.015029
Ankara/ <i>location</i> boundary/ <i>yes</i> Üniversitesi/ <i>organization</i>	0.000004

Table 1: The effect of the boundary flag on the performance of the tagger.

Output Sequence	Probability
Dr./ <i>else</i> boundary/ <i>yes</i> unk/ <i>person</i>	0.990119
Dr./ <i>else</i> boundary/ <i>yes</i> unk/ <i>location</i>	0.000690
Dr./ <i>else</i> boundary/ <i>yes</i> unk/ <i>org.</i>	0.000880
Dr./ <i>else</i> boundary/ <i>else</i> unk/ <i>else</i>	0.002688

Table 2: The use of the contextual model for unknown words. *org.* denotes “organization”.

to tag this word by replacing it with the flag “unk”, and let the model choose for the maximum probable tag considering the neighboring word “Dr.”. Table 2 gives the probabilities of the output sequences in which “Tür” is tagged as person, location, organization, or else, assuming that “Dr.” is not a part of the name.

More formally, this model helps tagging unknown words by modeling the following 4 cues:

1. Previous token in the same name, e.g. First names of the persons in a context like “Gökhan Tür”, assuming that first names constitute a smaller set than the surnames,
2. Previous token outside of the name, e.g. “Mr.”, “Dr.”, “Sayın” (Dear), in a context like “Sayın Tür”,
3. Next token in the same name, e.g. “Üniversitesi” (University), “Hastanesi” (Hospital), in a context like “Manitoba Üniversitesi” (Manitoba University),
4. Next token outside of the name, e.g. “de” (locative case marker), “kentinde” (in the city), in a context like “İzmir’de” (in Smyrna), or “İzmir kentinde” (in the city of Smyrna).

These cues can be considered analogous to the prepositions in English. Since, Turkish is an agglutinative language, there are no such prepositions, but the corresponding suffixes are attached to the words. If the word is a proper name, the word and the suffix are separated using an apostrophe. We considered these suffixes after the apostrophe as separate tokens, and this helped us a lot in contextual modeling.

3.3. Morphological Model

For morphological modeling, we morphologically analyzed all the words in our training data using the analyzer developed by Oflazer (1993), disambiguated them using the statistical morphological disambiguator of Hakkani-Tür (2000 forthcoming), and used the morphological parses of the words while training, instead of the surface forms.

We also added case information to the morphological parses, to indicate whether:

- the word is all in lower case, (NOCAP), e.g. “ev” (house),
- the word is all in upper case, (ALLCAP), e.g. “CNN”, or
- only the initial letter of the word is in upper case, (CAP), e.g. “Demirel”. For this case, we did not mark whether it is sentence initial or not.

We expected the morphological analyses of the words would help us in two ways:

1. Our morphological analyzer has a proper name database, and marks common Turkish person, location, and organization names as proper. In the morphological model, we can expect words, marked as proper are also to be marked as names.
2. The names are mostly noun phrases, and during training, we can expect the morphological model to learn such patterns. For example consecutive two proper nouns is a common person pattern, as in “George Washington”.

Since the lexicon of our morphological analyzer does not distinguish proper nouns with respect to their types, and there is no other way for this model to distinguish different names syntactically, morphological model only decides whether a word is a name or not. While tagging using only morphological model, we tag the words marked as name with the most popular name type, i.e. “person”. While combining this model with other models, we give the same probability to all of the name types.

Let’s demonstrate these expectations using a concrete example. Similar to Tables 1 and 2, Table 3 gives the probabilities for the named entity:

```
<ENAMEX TYPE="PERSON"> Süleyman Demirel
</ENAMEX>
```

where, both “Süleyman” and “Demirel” are analyzed as:¹

“Noun+Prop+A3sg+Pnon+Nom+CAP”.

3.4. Tag Model

The tag model is a trigram language model, which does not include any lexical items, but only the name tags, i.e. person, location, organization, and else, and the boundary flag types, i.e., yes, no, and mid. So its vocabulary consists of these 7 tokens. We built it by extracting the lexical words in our training data, and leaving only these tags.

¹The morphological features used in this example are: *Noun* : Noun, *Prop* : Proper Name, *A3sg* : Third person singular agreement, *Pnon* : No possessive agreement, *Nom* : Nominative case

Output Sequence	Probability
Noun+Prop+A3sg+Pnon+Nom+CAP/person boundary/mid	0.300339
Noun+Prop+A3sg+Pnon+Nom+CAP/person	
Noun+Prop+A3sg+Pnon+Nom+CAP/else boundary/no	0.0231911
Noun+Prop+A3sg+Pnon+Nom+CAP/else	

Table 3: The use of the morphological model.

Output Sequence	Probability
person mid person	0.999870
person yes person	0.006076

Table 4: The use of the tag model.

When we analyzed the errors of our name tagger, we found out that, some multi-token names were separated into different names of same or different types. For example the name

<ENAMEX TYPE="PERSON"> Alaattin Eroğlu
</ENAMEX>

was incorrectly tagged as

<ENAMEX TYPE="PERSON">Alaattin</ENAMEX>
<ENAMEX TYPE="PERSON">Eroğlu</ENAMEX>

Such a tagging damages the performance in two ways:

1. One of the names is marked as “spurious” by the evaluation software.
2. The other one’s “type” is correct, but “text” is marked as wrong².

On the other hand, the tag models favor for the correct tagging as seen in Table 4.

In other words, the function of this model is to limit the unprobable tag sequences, rather than finding names. Thus, we can expect the number of spurious and incomplete tags in our output to decrease, hence our performance to increase.

3.5. Model Combination

It is possible to tag a text using the lexical model or the morphological model alone. This is not the case for the other two models. Since morphological model does not include any lexical information, we do not expect the performance of the tagger to be high using only this model.

In order to tag using only lexical model, we set the state observation likelihoods to 1, and use only the lexical model in Viterbi decoding. Similarly, in order to tag using the morphological model, we first convert the tokens into their morphological parses, and use Viterbi decoding, then re-convert them into their original forms.

²See Section 4.2. for a detailed explanation of the evaluation metrics.

In order to combine the lexical model with the contextual model, we simply weighted interpolated these two models. The optimum weight is chosen using a separate held-out set. This mixture model can then be used in Viterbi decoding.

Combining lexical model and the morphological model is not that easy. Instead of interpolating the models, we have to interpolate the posterior probabilities, since one uses lexical forms of the words, while the other uses the morphological parses. We interpolated the posterior probabilities using empirically optimized weighting using a separate held-out set. After this interpolation, we can select the most probable tag for each word.

More formally, using lexical model, we can compute:

$$P_{LM}(w_i/t_i|w_{i-2}/t_{i-2}, w_{i-1}/t_{i-1})$$

where LM denotes lexical model, w_i denotes the i^{th} word (this can be either a real word, or a boundary), and t_i denotes the tag of that word.

Using our HMM, we can also compute the posterior

$$P_{LM}(w_i/t_i|w_{i-2}, w_{i-1})$$

$P(w_i/t_i) = P(t_i/w_i)$, since w_i is given. Hence, we can rewrite the above posterior as follows:

$$P_{LM}(t_i|w_{i-2}, w_{i-1}, w_i)$$

Similar to this notation, the morphological model can give us the posterior:

$$P_{MM}(M(w_i)/t_i|M(w_{i-2})/t_{i-2}, M(w_{i-1})/t_{i-1})$$

where MM denotes morphological model, $M(w)$ denotes the morphological analysis of the word w . Following the above notation we can say that this posterior is equal to:

$$P_{MM}(t_i|M(w_{i-2}), M(w_{i-1}), M(w_i))$$

Then, we can simply interpolate these posteriors with some weight λ as follows.

$$P_{LM+MM}(T|W, M(W)) = \lambda P_{LM}(T|W) + (1 - \lambda) P_{MM}(T|M(W))$$

where T denotes the sequence of tags, t_i , W denotes the input string, $M(W)$ denotes the morphological analyses of the words in the input string, $M(w_i)$.

Combining the morphological model with the mixture of the lexical and the contextual models can also be possible

by interpolating the posterior probabilities obtained from these information sources. The formal equations for this combination are very similar to combining morphological and lexical models.

Up to this point the tag model is not used in the combinations. In fact, the use of the tag model needs a little trick. In order to use this model, we used the posterior probabilities obtained from any combination of the other three models as state observation likelihoods, and use the tag model in order to determine the transition probabilities. One problem with this operation is converting posteriors, $P(T|W)$, to likelihoods, $P(W|T)$. This conversion is possible using the Bayes' rule:

$$P(W|T) = \frac{P(T|W)P(W)}{P(T)}$$

Since we try to optimize the output sequence, and $P(W)$ is given, hence constant, division of the posteriors to priors is proportional to the likelihood, and can be used in Viterbi decoding. In this HMM, the transition probabilities can be obtained using the tag model.

Combining all models can be stated more formally as follows:

$$P_{LM+MM+CM+TM}(T|W, C(W), M(W), T(W)) \propto P_{LM+CM+MM}(T|W, C(W), M(W)) \times P_{TM}(T)/P(T)$$

where CM denotes contextual model using contexts of the words, $C(W)$, TM denotes tag model using the tag sequence $T(W)$, λ is an empirically determined balancing parameter to adjust the dynamic ranges of the combined models.

Figure 5 shows a set of possible combinations of four models. Note that, there are also other ways of combining these models. For example, it is possible to combine lexical and tag models, by obtaining the posteriors from the lexical model, convert to likelihoods, and decode using the tag model as transition probabilities.

4. Experiments and Results

In this section, we report the results of evaluating the Turkish name tagger using the MUC evaluation software. In order to better understand the power of the models, and their combinations, we also present results for tagging English, using same models and evaluation metrics.

4.1. Training and Test Data

We trained our system using 492,821 words of newspaper articles containing 16,335 person names, 11,743 location names, and 9,199 organization names, summing up to 37,277 names. For testing we used about 28,000 words of newspaper articles, containing 924 person names, 696 location names, and 577 organization names, summing up to 2,197 names.

4.2. Evaluation Metrics

Along with the definition of the named entity extraction task, the evaluation metrics are also set by the MUC program. MUC scoring software is used to evaluate the systems participated in these conferences.

For the name tagging task, there are 2 criteria to evaluate:

- **Type:** Checks for the type of the name, i.e. person, location, or organization.
- **Text:** Checks for the text marked as a name.

For each of these 2 criteria, the evaluation software computes the following 3 values:

- **Correct:** Number of correct answers found by the name-finder.
- **Actual:** Number of answers found by the name-finder.
- **Possible:** Number of possible correct answers in the key.

For *Type* and *Text* criteria, the above 3 values are summed up. Then, two metrics borrowed from the information retrieval community, *recall* and *precision* values are computed as follows:

$$Recall = \frac{Correct\ Type + Correct\ Text}{Possible\ Type + Possible\ Text}$$

$$Precision = \frac{Correct\ Type + Correct\ Text}{Actual\ Type + Actual\ Text}$$

Informally, recall measures the number of hits vs. the number of possible correct answers as specified in the key, whereas precision measures how many answers were correct ones compared to the number of answers delivered. There is no partial credit in *Text* criterion. Even though most of the words of a name have been marked, this is called as incorrect.

Finally, these two measures of performance are combined to form one measure of performance, the *F-measure*, which is computed by the uniformly weighted harmonic mean of precision and recall:

$$F - measure = \frac{Recall \times Precision}{\frac{1}{2} \times (Recall + Precision)}$$

4.3. Results

Table 1 gives the accuracy of our system according to the MUC evaluation metrics. We have provided results using only lexical and morphological information in addition to the 4 types of combinations shown in the table, although it is possible to combine these information sources in 11 different ways. In all of the combinations, we did not separate the lexical model from the contextual model, because lexical model alone is relatively very weak in tagging. So we are left with only 4 types of combinations.

We are very pleased to see that, lexical model alone performed high 80s. When we look at this model in detail, we see that we have done well in detecting the types of the names, but we have problems in detecting them. The main reason of this problem is the unknown words. This problem is solved by the contextual model, and the performance of the "Text" metric is increased to 86%. It is also interesting to see that the morphological model alone has performed

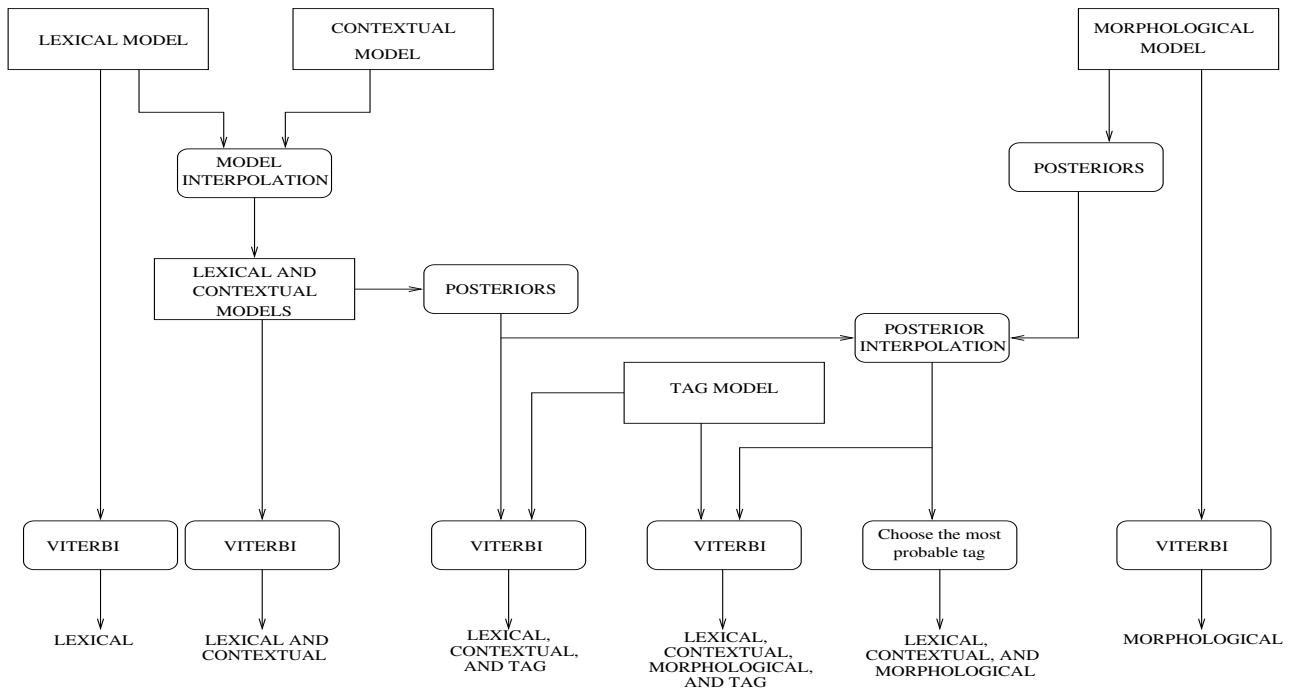


Figure 5: Combining lexical, contextual, morphological, and tag models for tagging Turkish text.

Model	Text	Type	F-Measure
Lexical	80.87%	91.15%	86.01%
Morphological	36.52%	79.73%	58.12%
Lexical+Contextual	86.00%	91.72%	88.86%
Lexical+Contextual+Morphological	87.12%	92.20%	89.66%
Lexical+Contextual+Tag	89.54%	92.13%	90.84%
Lexical+Contextual+Morphological+Tag	90.40%	92.73%	91.56%

Table 5: Accuracy of the name tagging task using lexical, contextual, morphological, and tag models.

about 58%, without even knowing the surface forms or the roots of the words, a score which was not expected even by us. We were also successful in incorporating the extra information held by the morphological model to the combination of lexical and contextual models, and gained 0.8% points more. Instead of the morphological model, when we have incorporated the tag model, we have gained about 2% points more. These improvements are important, since we have entered a range, in which it is very hard to achieve further improvements. Finally, when we have combined all of our models, we have reached 91.56%. We see that tag model is very effective in this task. The “Text” metric is increased more than 3% points, and “Type” metric is increased about 0.5% points in either cases this model was used. Similarly, the morphological model increases the F-measure by 0.8% in either cases it was used. When we compare the final F-Measure with our baseline lexical performance, we see an improvement of 5.55% points.

4.4. Error Analysis

Table 6 shows the performance of our name tagger with respect to name types. These are the results when we use all four of our models.

We see that our performance varies greatly for different

	Possible	Actual	Correct	F-Measure
Person	927	945	867	92.63%
Location	698	716	674	95.33%
Org.	576	607	531	89.77%
TOTAL	2201	2268	2072	92.73%

Table 6: Detailed name tagging results when we use all the models. `ORG` denotes the type “Organization”.

name types. It is also interesting to see that, our performance is best for locations, and worst for organizations. When we analyze our test data we see that our system performs not so satisfactory for very long organization names. For example the organization:

`<ENAMEX TYPE="ORGANIZATION">Adana Emniyet Müdürlüğü Organize Suç ve Silah Kaçakçılığı Şube Müdürlüğü</ENAMEX>`

was tagged as:

`<ENAMEX TYPE="ORGANIZATION">Adana Emniyet Müdürlüğü Organize Suç`

Language	Text	Type	F-Measure
Turkish	84.26%	90.72%	87.49%
English	82.95%	89.56%	86.26%

Table 8: Comparison of the Turkish and English name tagging results using only lexical and contextual models.

ve Silah</ENAMEX> Kaçakçılığı
<ENAMEX TYPE="ORGANIZATION">Şube
Müdürlüğü"</ENAMEX>

which results in two different names, neither of which were tagged as correct in "Text", and only one was tagged as correct in "Type".

4.5. Effect of the Case and Punctuation Information

Tagging Turkish becomes more critical when we remove case and punctuation information. Such an experiment is important in order to see the performance of the tagger with speech recognizer output (SNOR) format, which is largely unpunctuated (apostrophes are preserved) and in all capital letters, as set by the NIST (1998). Case is a very valuable information source in tagging proper names. Similarly punctuation has importance for this task in detecting the name boundaries, since most of the time, punctuation resolves ambiguities, such as "Mesut, Yılmaz" vs. "Mesut Yılmaz". While removing the punctuation, we did not touch the apostrophe sign, since only this punctuation is provided with the speech recognizer output. This nuance has extra importance for tagging Turkish, because we have been modeling a proper name and its suffixes separately. This is easy, because the apostrophe sign marks the boundary between the root and the suffixes. If we had to remove all the punctuations, we would lose this information, and we would expect to face with data sparseness in building our model.

In these experiments, our aim is not to improve our performance with input lacking case or punctuation, but instead to see our performance without any modification to the models and system.

Table 7 shows our results when we remove the case and/or punctuation information. We see that our models can still be used in such a case since we did suffer too much. The decrease in the performance was 2.76% when using lexical, contextual, and tag models. Indeed, these results are comparable with the results of BBN (Bikel et al., 1999). They have reported a loss of 4.2% (from 94.9% to 90.7%) on the Wall Street Journal articles using the SNOR format.

4.6. Results Compared to Name Tagging of English

In order to see whether these results are comparable with the results obtained for English, we built a similar system using similar statistical methods. Table 8 presents the performance of our algorithm applied to both English and Turkish input in SNOR format.

5. Conclusion

We presented a probabilistic model for automatically tagging names in a Turkish text. We used four different information sources to model names, and successfully combined them. Our first information source is based on the surface forms of the words. Then we combined the contextual clues, and obtained a significant win. After this, we modeled the morphological analyses of the words, and reached an F-measure of 89.66% according to the MUC evaluation software, which was 3.65% points better than the lexical model alone. Finally, we modeled the tag sequence, and gained 1.90% more, reaching an F-measure of 91.56% in Turkish name tagging.

This was the second study on Turkish named entity extraction. Cucerzan and Yarowsky (1999) reported an F-measure of 53% using very little training data. This implies the importance of the size of the training data in corpus based language processing tasks. The huge difference in the training data sizes makes a comparison impossible. Instead, we gave results for English, using the same lexical and contextual models, and see that it is possible to reach an F-measure of 86%.

These results are important in the following senses:

- We have successfully combined lexical, contextual, morphological, and tag information for this basic information extraction task. Each model contributed to this task as expected.
- Using the lexical model alone performed in high 80s for Turkish name tagging, which is a very similar result we obtained for English. Thus, we can claim that statistical methods can be used for name tagging task even for agglutinative languages.
- We have seen that removing the case information results in a 3% points decrease in F-measure. This implies that, statistical methods can still be used for name tagging of speech.
- Recalling that named entity extraction task is a prerequisite task for other more complex information extraction tasks, we are now ready to move on other tasks.
- As a future research, we would like to work on the problems we have encountered especially in tagging organizations.

6. Acknowledgments

This work was begun while the first two authors were visiting Speech Technology and Research Laboratory, SRI International, with support from DARPA under contract no. N66001-97-C-8544 and from NSF under grant IRI-9619921. We thank Andreas Stolcke and Elizabeth Shriberg for many helpful discussions.

7. References

- J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. 1995. MITRE: Description of the Alembic system used for MUC-6. In *Proceedings of the MUC-6*.

<i>Model</i>	<i>Text</i>	<i>Type</i>	<i>F-Measure</i>
Lexical	80.71%	90.17%	85.44%
Lexical+Contextual	84.26%	90.72%	87.49%
Lexical+Contextual+Tag	90.88%	85.28%	88.08%

Table 7: Accuracy of the name tagging task using lexical, contextual, and tag models without case and punctuation information.

- S. W. Bennett, C. Aone, and C. Lovell. 1997. Learning to Tag Multilingual Texts through Observation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, 34:211–231. Special Issue on Natural Language Learning.
- A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998a. NYU: Description of the MENE Named Entity System as Used in MUC-7. In *Proceedings of the MUC-7*.
- A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998b. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings 6th Workshop on Very Large Corpora at the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Canada, August.
- Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, Department of Computer Science, University of Pennsylvania.
- N. Chinchor and P. Robinson. 1998. MUC-7 Named Entity Task Definition (version 3.5). In *Proceedings of the MUC-7*.
- Silviu Cucerzan and David Yarowsky. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proceedings of the Workshop on Very Large Corpora at the Conference on Empirical Methods in NLP*.
- I. J. Good. 1953. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40:237–264, December.
- Dilek Hakkani-Tür, Gökhan Tür, Andreas Stolcke, and Elizabeth Shriberg. 1999. Combining Words and Prosody for Information Extraction from Speech. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, volume 5, pages 1991–1994, Budapest, September.
- Dilek Z. Hakkani-Tür. 2000, forthcoming. *Statistical Language Modeling for Turkish*. Ph.D. thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey.
- J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. 1996. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Roche and Shabes, editors, *Finite State Devices for Natural Language Processing*, pages 383–406. MIT Press, Cambridge, MA.
- S. Katz. 1997. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401, March.
- G. R. Krupka and K. Hausman. 1998. IsoQuest Inc.: Description of the *NetOwlTM* Extractor System as Used for MUC-7. In *Proceedings of the MUC-7*.
- D. Lin. 1998. Using Collocation Statistics in Information Extraction. In *Proceedings of the MUC-7*.
- A. Mikheev, C. Grover, and M. Moen. 1998. Description of the LTG System Used for MUC-7. In *Proceedings of the MUC-7*.
- Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and the Annotation Group. 1998. Algorithms that Learn to Extract Information; BBN: Description of the SIFT System as Used for MUC-7. In *Proceedings of the MUC-7*.
- MUC-6, 1995. *Proceedings of the MUC-6*.
- MUC-7, 1998. *Proceedings of the MUC-7*.
- NIST, National Institute of Standards and Technology, 1998. *The 1998 Hub-4 Evaluation Plan for Recognition of Broadcast News, in English*.
- Kemal Oflazer. 1993. Two-level Description of Turkish Morphology. *Literary and Linguistic Computing*, 8(3).
- J. R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning*, 1(1):81–106.
- Andreas Stolcke. 1999. SRILM—the SRI language modeling toolkit. <http://www.speech.sri.com/projects/srilm/>.
- Gökhan Tür. 2000, forthcoming. *A Statistical Information Extraction System for Turkish*. Ph.D. thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey.
- A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.