Natural Language Engineering **9** (2): 181–210.
DOI: 10.1017/S135132490200284X Printed in the United Kingdom

A statistical information extraction system for Turkish[1]

GÖKHAN TÜR, DILEK HAKKANI-TÜR
AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932, USA

KEMAL OFLAZER
Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul TR-81474, Turkey

(*Received 26 January 2001; revised 12 October 2001*)

Abstract

This paper presents the results of a study on information extraction from unrestricted Turkish text using statistical language processing methods. In languages like English, there is a very small number of possible word forms with a given root word. However, languages like Turkish have very productive agglutinative morphology. Thus, it is an issue to build statistical models for specific tasks using the surface forms of the words, mainly because of the data sparseness problem. In order to alleviate this problem, we used additional syntactic information, i.e. the morphological structure of the words. We have successfully applied statistical methods using both the lexical and morphological information to sentence segmentation, topic segmentation, and name tagging tasks. For sentence segmentation, we have modeled the final inflectional groups of the words and combined it with the lexical model, and decreased the error rate to 4.34%, which is 21% better than the result obtained using only the surface forms of the words. For topic segmentation, stems of the words (especially nouns) have been found to be more effective than using the surface forms of the words and we have achieved 10.90% segmentation error rate on our test set according to the weighted TDT-2 segmentation cost metric. This is 32% better than the word-based baseline model. For name tagging, we used four different information sources to model names. 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 some 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%, according to the MUC evaluation criteria. 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 significantly improves these basic information extraction tasks for Turkish.

1 Introduction

This paper presents the results of a study on information extraction from unrestricted Turkish text using statistical language processing methods. Information Extraction

[1] This work was done while the first and second authors were PhD students at Bilkent University, Ankara, Turkey.

Table 1. *Comparison of the number of unique word forms in English and Turkish, in large text corpora*

Language	Vocabulary Size
English	97,734
Turkish	474,957
Turkish (only roots)	94,235

(IE) is the task of extracting particular types of entities, relations, or events from natural language text or speech. The notion of what constitutes information extraction has been heavily influenced by the *Message Understanding Conferences* (MUCs) (MUC 1995; MUC 1998; Grishman 1998; Grishman and Sundheim 1996). This conference has been extended also to handle other languages, such as Spanish, Japanese and Chinese in the Multilingual Entity Task (MET) conferences. A relatively new conference also related to information extraction is the *Topic Detection and Tracking Conference* (TDTs) which refers to automatic techniques for finding topically related material in streams of data (e.g. newswire and broadcast news) (Wayne 1998).[1]

In text and speech processing, the availability of more data and more tools has recently motivated the use of statistical methods. For example, we used the SRILM toolkit for language modeling and decoding in this work (Stolcke 1999). Although such methods have long been used in the speech community, it became popular in the late 1980s, and early 1990s in natural language processing tasks, such as machine translation (Brown, Cocke, Della Piettra, Della Piettra, Jelinek, Lafferty, Mercer and Roossin 1990), part of speech tagging (Church 1988) and information extraction (Bikel, Schwartz and Weischedel 1999).

In Turkish, using the surface forms of the words results in data sparseness in the training data. Table 1 shows the size of the vocabulary obtained by a recent study conducted by (Hakkani-Tür 2000) on corpora of Turkish and English, of about 10 million words, collected from online newspapers.

The main reason for this discrepancy is that, Turkish word formation has very productive inflectional and derivational processes, where it is possible to produce thousands of forms (or even millions (Hankamer 1989)) for a given root word. Note that, the size of the vocabulary decreased on the order of 5, when we use the roots of the words, and became comparable with English. This data sparseness poses a challenging problem for statistical language processing, and we think the most effective way to handle this is to exploit the morphological analyses of the words.

For instance, the derived modifier *sağlamlaştırdığımızdaki* (Literally, "(the thing existing) at the time we caused (something) to become strong") would be morphologically decomposed as:

sağlam+laş+tır+dı+ğı+mız+da+ki

[1] For more information on these tasks, see (Tür 2000).

and morphologically analyzed as:[2]

```
sağlam+Adj ^DB
+Verb+Become ^DB
+Verb+Caus+Pos ^DB
+Adj+PastPart+P1sg ^DB
+Noun+Zero+A3sg+Pnon+Loc ^DB
+Adj
```

A Turkish word can be represented as a sequence of *inflectional groups* (IGs) as described by Oflazer (1999). An IG is a sequence of inflectional morphemes, separated by derivation boundaries (^DB). For example, the above word, *sağlamlaştırdığımızdaki*, would be represented with the following six IGs:

1. sağlam+Adj
2. Verb+Become
3. Verb+Caus+Pos
4. Adj+PastPart+P1sg
5. Noun+Zero+A3sg+Pnon+Loc
6. Adj

Note that lexicalized derivations such as "gözlem" (observation) appear as distinct words. IGs only separate productive derivations whose lexical semantics are pretty much predictable. In cases where a word is both lexicalized and also corresponds to a productive derivation, both analyses are produced by the morphological analysis.

Any overt or covert morphological phenomenon that is considered to cause a derivation marks an IG boundary. These phenomena include (by convention) for instance phenomena like passivization, causativization of verbs, or semantic modifications by modal suffixes (of which – "yabil" is one example – there are about 10). IGs are marked by a new part of speech (which may be the same as the old part of speech) and a minor part of speech typically marking a finer distinction, and any further inflectional phenomena associated with the relevant derived form. An interesting observation that we can make about Turkish is that, when a word is considered as a sequence of IGs, syntactic relation links only emanate from the last IG of a (dependent) word, and land on one of the IGs of the (head) word on the right. Thus intermediate IGs may have relevance in modeling relationships (Hakkani-Tür and Oflazer 2000).

We have used the final IGs of the words in name tagging and topic segmentation tasks, for the following reasons:

- The final IG determines the final category, hence its function of a word. For example, our example word is unlikely to be a sentence final word, since its final category is adjective. Recall that Turkish is a head-final language, i.e. sentences generally end with a finite verb.

[2] The morphological features used in this word are given in Appendix A.

Table 2. *Numbers of analyses and IGs in Turkish*

	Possible	Observed
Full analyses (No roots)	∞	10,531
Inflectional groups	9,129	2,194

- The use of the final IG instead of the whole morphological analysis solves the problem of data sparseness. While there may be theoretically infinitely many such word forms in Turkish, the number of possible final IGs is limited. Table 2 presents the number of IGs observed in a corpus of 1 million words (Hakkani-Tür 2000).

2 Sentence segmentation

Sentence segmentation is the task of automatically dividing a stream of text or speech into grammatical sentences. Given a sequence of (written or spoken) words, the aim of sentence segmentation is to find the boundaries of the sentences. Sentence segmentation is a preliminary step towards speech understanding. Many natural language and speech processing tasks, such as parsing the sentence, finding topic changes, aligning multilingual text, require their input to be divided into sentences. Once the sentence boundaries have been detected, then further syntactic and/or semantic analysis can be performed on these sentences. Furthermore, if you deal with speech recognizer output, it lacks the usual textual cues to these entities (such as headers, paragraphs, sentence punctuation, and capitalization).

In our previous work at SRI International, STAR Lab., for English, we tried to combine the lexical model with the prosodic model (Stolcke, Shriberg, Bates, Ostendorf, Hakkani, Plauché, Tür and Lu 1998; Stolcke, Shriberg, Hakkani-Tür, Tür, Rivlin and Sönmez 1999; Shriberg, Stolcke, Hakkani-Tür and Tür 2000; Hakkani-Tür, Tür, Stolcke and Shriberg 1999). Lexical information was modeled using an *n*-gram language model, trained from 130 million annotated words. Besides this lexical model, we built a separate prosodic model using a decision tree which classifies the word boundaries as sentence boundary or non-sentence boundary. This prosodic model was trained using the prosodic features obtained from 700,000 words of broadcast news transcripts.

Note that, this task differs from the works of Reynar and Ratnaparkhi (1997) and Palmer and Hearst (1997), where they use punctuation information to segment the sentences.

2.1 Approach

Like all other tasks described in this paper, we used a statistical model for this task. We grouped the words into contiguous stretches belonging to one sentence, i.e. the word boundaries were classified into "sentence boundaries" and "non-sentence

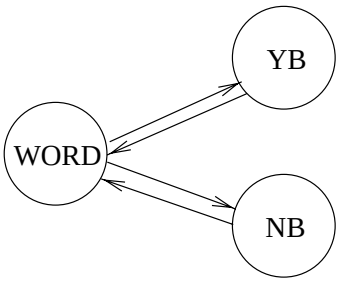

Fig. 1. The conceptual figure of the HMM used by SRI for sentence segmentation. YB denotes that there is a sentence boundary, NB denotes that there is no sentence boundary, $WORD$ denotes the words of the text.

boundaries". The sentence segmentation task was thus reduced to a boundary classification problem. We will use B to denote the string of binary boundary classifications, and W to denote the word sequence. Our approach aimed to find the segmentation B with highest probability given the information in W.

$$\underset{B}{\operatorname{argmax}}\, P(B|W)$$

We formed our training data, so that each word was followed by a boundary flag which denotes whether there was a sentence boundary or not. For example, the word sequence:

... çoğalmayı sağladı <**S**> ne olduysa oktay'ın ...

was converted to

... NB çoğalmayı NB sağladı YB ne NB olduysa NB oktay'ın NB ...

where YB and <S> denote that there is a boundary, and NB denotes otherwise. As our input lacked punctuation, case, or other acoustic and prosodic information, we had no source of information other than words. We have made use of the surface forms and morphological analyses of the words.

Similar to our work at SRI, we built an HMM, as depicted in Figure 1, in which states either denote whether there was a boundary or not between two words, or denote the words in the text. Transition probabilities were obtained from a language model.

2.1.1 Word-based model

We built a language model using only surface forms of the words similar to the SRI system. This model also enabled us to gauge our baseline performance.

The language model was formed from the training data, as described above. In order to see the effect of this model, consider the portion "... geldi çünkü ..." (... came because ...). Table 3 shows the probabilities of the possible taggings for this

Table 3. *The effect of the word-based language model. In this example, using only words, we can say that it is 30 times more probable to have a sentence boundary in-between*

Output sequence	Probability
geldi *NB* çünkü	0.00028166
geldi *YB* çünkü	0.00614714

Table 4. *The effect of the word-based language model*

Output sequence	Probability
Verb+Pos+Past+A3sg *NB*	0.24849
Verb+Pos+Past+A3sg *YB*	0.751505

text piece.[3] As seen, the one with the boundary has about 30 times more probable than the other.

2.1.2 Morphological model

In addition to the surface forms of the words, we used the morphological analyses of the words, which hold valuable information for this task, and alleviate the data sparseness problem we would encounter in building the language model.

While forming the morphological model, we used the final inflectional groups of the morphological analyses of the words instead of the surface forms.

To build the morphological model, we used a preprocessing module, developed by Hakkani-Tür (2000), which tokenizes the training data, analyzes the tokens using the morphological analyzer developed by Oflazer (1993), groups the collocations, removes some obviously improbable morphological parses in order to reduce the morphological ambiguity, and finally gives the most probable morphological analyses, corresponding to the words.

Table 4 shows the probabilities of the possible taggings after the word "geldi" (came) according to the morphological model. The word "geldi" is morphologically analyzed as "*gel+Verb+Pos+Past+A3sg*".[4] As seen, it is about three times more probable of marking a sentence boundary after a final verb.

[3] The probabilities in all the tables are posterior probabilities.
[4] See Appendix A for the meanings of these features in the morphological analyses.

2.1.3 Model combination

We preferred the posterior probability interpolation method, in order to combine these two information sources.

$$P(T|W, M(W)) \approx \lambda P_{\mathrm{WM}}(T|W) + (1 - \lambda) P_{\mathrm{MM}}(T|M(W))$$

where WM denotes the word-based model, MM denotes the morphological model, T denotes the boundary type, W denotes the word sequence, $M(W)$ denotes the sequence of the last IGs of W as produced by the morphological analyzer and disambiguator. λ is a parameter optimized on held-out data to optimize the overall model performance.

2.2 Experiments and results

To evaluate the word-based and morphological model, and their combined performance, we carried out experiments described in this section. We first describe our training and test data, then give results obtained with the word-based, morphological language models and their combinations.

2.2.1 Training and test data

The word-based model was trained using the web resources of Milliyet newspaper articles, covering the period from January 1 1997 through September 12 1998, containing about 18 million words, 50,674 sentences. To see the effect of the training size, we also used a small subset of 1 million words from this corpus for training. Test data contains 14,738 words, 931 sentences from the same newspaper.

2.2.2 Evaluation metrics

According to our evaluation metric, each word boundary is marked as sentence or non-sentence boundary, and we align the output with manually annotated test data, and finally compute the error rate with the following formula:

$$\text{Error Rate} = \frac{\text{Number of False Alarms} + \text{Number of Misses}}{\text{Number of Boundaries}}$$

2.2.3 Sentence segmentation results

Table 5 shows our performance using word-based and morphological models, and their combinations. The chance performance for this task is 8.65% error, obtained by labeling all locations as non-sentence boundaries (the most frequent class). The baseline performance was obtained by marking all word boundaries, where the preceding words' morphological analyses contain the finite Verb tag in their final IGs. As sentence boundaries, we ignored imperative verbs, because such analyses occurred most of the time, if the word was mis-analyzed, or if it was in a quotation.

Results show that the morphological model alone performs better than a word-based language model, unless the language model was trained on a much larger

Table 5. *Results for Turkish sentence segmentation using word-based, morphological language models, and their combinations. LM denotes the word-based model, and MM denotes the morphological model.* Baseline *denotes the performance, when we put a sentence boundary after every finite verb*

Model	Error (%) rate
Chance	8.65
Baseline	5.85
LM only (1M words)	5.98
LM only (18M words)	4.82
MM only (1M words)	4.90
MM only (18M words)	4.90
LM (18M) + MM (1M)	4.59
LM (18M) + MM (18M)	4.34

data set. This is a typical result of the data sparseness we have encountered while training the word-based model. Training with 1 million words performed even worse than the baseline. An interesting result is that, the morphological model performed similarly when trained with 1 million and 18 million words, although this similarity disappeared interestingly when combined with the word-based model. Also it is worthwhile to note that it is possible to get close performances with the morphological model trained with 1 million words, instead of a word-based model trained with 18 times more data. Most importantly, error reductions of 21% and 25% over the baseline were achieved by combining the word-based model trained with 18 million words with the morphological model trained with 1 million and 18 million words consecutively.

2.2.4 Error analysis

When we analyzed our errors, we saw three main categories of errors:

1. The system sometimes made errors while deciding to end the sentence after the words, which could be used as final verb, or derived adjective. For example, the word "*düzenlenecek*" (literally "will be organized") is morphologically ambiguous. It can either be an adjective, or a verb. This is indeed due to the errors made by the morphological disambiguator.
2. Since we were dealing with newspaper articles, titles were also marked as sentences. It was very hard to determine the boundaries in such sentences.
3. According to our conventions, we did not mark sentence boundaries for the nested sentences inside a quoted piece of text. It was very hard without punctuation even for humans to decide sentence boundaries in such cases.

2.2.5 Results compared to sentence segmentation of English

When we compare our results with the ones obtained for English at SRI, we see that, using 130 million words for training, the error rate was 4.1% for this task. Considering the difference in the training data size, we can say that our results are comparable with English. Furthermore, we see that using syntactic information, such as part-of-speech information of the words, we can gain some points for English, and this can be a promising future research.

3 Topic segmentation

Topic segmentation is the task of automatically dividing a stream of text or speech into topically homogeneous blocks. Given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics change. Topic segmentation is an important task for various language understanding applications, such as information extraction and retrieval (IR), and text summarization. An application may be as follows: Given a corpus of newspaper articles strung together, and a user's query, return a collection of coherent segments matching the query. Lacking a tool for detecting topic breaks, an IR application may be able to locate positions in its database, but be unable to determine how much of the surrounding data to provide to the user (Beeferman, Berger and Lafferty 1999). Another example may be the broadcast news, or video-on-demand applications. There is no mark-up to indicate the topic boundaries and even the sentence boundaries in broadcast news. Also, segmenting text along topic boundaries may be useful for text summarization and anaphora resolution (Kozima 1993).

There has recently been increased interest in segmenting such information streams into topics. In 1997, the US Defense Advanced Research Projects Agency (DARPA) initiated the Topic Detection and Tracking (TDT) Program (Allan, Carbonell, Doddington, Yamron and Yang 1998). The purpose of this effort is to advance and accurately measure the state of the art in TDT and to assess the technical challenges to be overcome. In the framework of this program, a topic is defined to be a seminal event or activity, along with all directly related events and activities. Thus, topic segmentation is an *enabling* technology for other applications, such as topic tracking and new event detection. We have also followed this framework in the definition and evaluation of this task.

3.1 The approach

Similar to the sentence segmentation task, we tried to classify the sentence boundaries as "topic boundaries" and "nontopic boundaries".

For topic segmentation, we used an extension of Dragon's system (Yamron, Carp, Gillick, Lowe and van Mulbregt 1998; van Mulbregt, Carp, Gillick, Lowe and Yamron 1998). In this system, lexical information is captured by statistical language models (LMs) embedded in an HMM. We preserved the HMM structure, in which states correspond to topic clusters T_j and the observations are sentences $W_1, \ldots, W_N$, as given in Figure 2. In their scheme, the observation likelihoods for the HMM states, $P(W_i|T_j)$, are obtained from the corresponding topic cluster language models. This

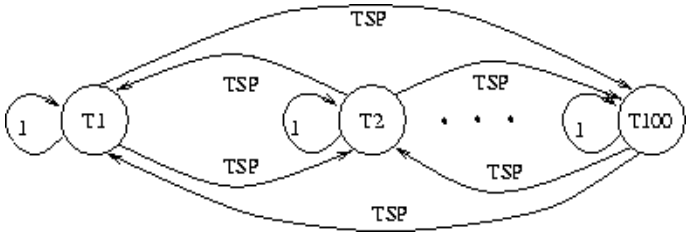

Fig. 2. Structure of the basic HMM developed by Dragon for the TDT Pilot Project. The labels on the arrows indicate the transition probabilities. TSP represents the topic switch penalty.

approach was based purely on topical word distributions. In our extension of this model, we incorporated morphological aspects of Turkish, using stems of the words and then using only nouns in forming the topic clusters, as described in the following subsections.

3.1.1 Word-based modeling

To gauge our baseline performance, similar to Dragon, we automatically constructed 100 topic cluster LMs, using the multipass *k*-means algorithm described in Yamron, Carp, Gillick, Lowe and van Mulbregt (1998). Since the HMM emissions are meant to model the topical usage of words, but not topic-specific syntactic structures, the LMs consist of unigram distributions that exclude stop words (high-frequency function and closed-class words). To account for unobserved words we interpolate the topic cluster-specific LMs with the global unigram LM obtained from the entire training data. The observation likelihoods of the HMM states are then computed from these smoothed unigram LMs.

Table 3.1.1 gives a list of the most frequent words in the same topic cluster, containing mostly soccer news articles. *Beşiktaş*, *Galatasaray*, *Fenerbahçe* and *Trabzonspor* are top Turkish soccer teams, *Hakan*, *Mehmet*, and *Ali* are the top players and *Fatih Terim* is the trainer of *Galatasaray*.

3.1.2 Stem-based modeling

Word-based modeling works well in languages in which there is very little or no morphology, such as English. On the other hand, morphologically rich languages, like Turkish, suffer from the data sparseness while using the surface forms of the words results in the training data. When we consider the words with different inflectional and derivational suffixes different, then we have to deal with data sparseness.

Table 3.1.2 gives a list of 26 different word forms involving the stem *gol* (goal), in the cluster mentioned in Table 3.1.1.

This sparseness does not only badly damage the quality of the language models, but also the performance of the clustering algorithm. Since we check for the similarity distance of a given document and a cluster, and use the words themselves in this computation, the result may be misleading while using the words. So we can expect a better clustering using stemming beforehand.

Table 6. *The most frequent words in one of the clusters, containing mostly soccer news articles.* Loc *denotes locative case,* Acc *denotes accusative case*

Word	Freq	Meaning
gol	1222	goal
ikinci	912	second
Beşiktaş	867	Beşiktaş
teknik	781	technical
Galatasaray	773	Galatasaray
Fenerbahçe	699	Fenerbahçe
orta	678	middle
takım	665	team
dk.	655	min.
sarı	622	yellow
maç	592	match
yarıda	575	half *Loc*
top	521	ball
Trabzonspor	479	Trabzonspor
yaptı	473	did
Mehmet	471	Mehmet
Hakan	462	Hakan
dakikada	450	minute *Loc*
maçı	449	match *Acc*
futbol	445	soccer
Fatih	413	Fatih
yarı	412	half
oyun	406	game
Ali	384	Ali

It is clear that, removing the suffixes the words, and using the root words will prevent the data sparseness, and the unigram language models obtained from the topic clusters would be more effective. So we decided to use the root words instead of the surface forms of the words, and build stem-based language models, instead of word-based language models.

In order to do this, we extracted the roots of the words, using the disambiguated morphological analyses (as done in the sentence segmentation task), and rebuilt the training corpus using only these roots. When there were more than one root for a word, because of the morphological ambiguity, we used all of the roots. However, this root ambiguity was not a real problem as there were only 1.15 distinct roots per word on the average.

As expected, we obtained clusters with smaller numbers of root words, and each with higher frequencies. Table 3.1.2 lists the most frequent root words in corresponding cluster containing mostly soccer news.

3.1.3 Noun-based modeling

When we analyzed Table 3.1.2, and other clusters, we saw that in order to model the topical usage of words, it was not enough to exclude the stopwords. In fact, only

Table 7. *The frequency table for the root word* gol *(goal) in the cluster mentioned in Table 3.1.1*

Word	Freq	Morphological Analysis
gol	1222	goal+Noun+A3sg+Pnon+Nom
golü	350	goal+Noun+A3sg+Pnon+Acc or goal+Noun+A3sg+P3sg+Nom
gole	150	goal+Noun+A3sg+Pnon+Dat
golle	138	goal+Noun+A3sg+Pnon+Ins
goller	126	goal+Noun+A3pl+Pnon+Nom
golde	85	goal+Noun+A3sg+Pnon+Loc
golün	75	goal+Noun+A3sg+Pnon+Gen or goal+Noun+A3sg+P2sg+Nom
golünü	63	goal+Noun+A3sg+P3sg+Acc or goal+Noun+A3sg+P2sg+Acc
golüyle	62	goal+Noun+A3sg+P3sg+Ins
golcü	59	goal+Noun+A3sg+Pnon+Nom ^DB+Adj+Agt
golleri	48	goal+Noun+A3pl+P3sg+Nom or goal+Noun+A3pl+Pnon+Acc or goal+Noun+A3pl+P3pl+Nom or goal+Noun+A3sg+P3pl+Nom
golden	45	goal+Noun+A3sg+Pnon+Abl
gollerle	40	goal+Noun+A3pl+Pnon+Ins
gollük	37	goal+Noun+A3sg+Pnon+Nom ^DB+Adj+FitFor
gollü	26	goal+Noun+A3sg+Pnon+Nom ^DB+Adj+With
golüne	24	goal+Noun+A3sg+P3sg+Dat or goal+Noun+A3sg+P2sg+Dat
golleriyle	20	goal+Noun+A3pl+P3sg+Ins or goal+Noun+A3pl+P3pl+Ins or goal+Noun+A3sg+P3pl+Ins
golsüz	18	goal+Noun+A3sg+Pnon+Nom ^DB+Adj+Without
golcüsü	18	goal+Noun+A3sg+Pnon+Nom ^DB+Noun+Agt+A3sg+P3sg+Nom
golünde	16	goal+Noun+A3sg+P3sg+Loc or goal+Noun+A3sg+P2sg+Loc
gollerde	15	goal+Noun+A3pl+Pnon+Loc
goldeki	15	goal+Noun+A3sg+Pnon+Loc ^DB+Det
gollerin	12	goal+Noun+A3pl+Pnon+Gen or goal+Noun+A3pl+P2sg+Nom
golünden	10	goal+Noun+A3sg+P3sg+Abl or goal+Noun+A3sg+P2sg+Abl
gollerini	9	goal+Noun+A3pl+P3sg+Acc or goal+Noun+A3pl+P2sg+Acc or goal+Noun+A3pl+P3pl+Acc or goal+Noun+A3sg+P3pl+Acc
gollere	8	goal+Noun+A3pl+Pnon+Dat

nouns would be sufficient to model the topics. Since we have the morphological analyses of the words, it was straightforward for us to test this hypothesis.

Instead of using the stems of words, we only used the stems of the morphological parses that have a noun root form. After using the same clustering algorithm, we ended up with new clusters. The most frequent nouns for the cluster containing mostly soccer related articles is listed in Table 3.1.3. Common verbs such as, *ol* (be), *al* (take), *yap* (make), and *et* (do) and somewhat soccer related verbs, such as *oyna*

Table 8. *The most frequent stems in a cluster, containing mostly soccer news articles*

Word	Freq	Meaning
gol	2271	goal
maç	2048	match
oyun	1781	game
takım	1382	team
ol	1317	be
oy[5]	1273	vote
al	1264	take
top	1228	ball
futbol	1227	soccer
oyna	1224	play
yap	1219	do or make
yarı	1101	half
Galatasaray	1018	Galatasaray
saha	996	field
Hakan	986	Hakan
Beşiktaş	974	Beşiktaş
at	948	throw
dakika	892	minute
Fenerbahçe	872	Fenerbahçe
rakip	866	opponent
çık	826	exit
orta	785	middle
et	755	do or make
ikinci	734	second

(play), *çık* (exit), and *at* (score) disappeared in Table 3.1.3 when we compare with Table 3.1.2.

3.2 Experiments and results

To evaluate our topic segmentation models we carried out experiments in the TDT paradigm. We first describe our training and test data, then give results obtained with the baseline word-based, stem-based, and noun-based language models. We assumed that each news piece contains only one topic, and attempted to find out article boundaries. Hand-checking of a subset of articles showed that this assumption was true except for a few cases.

3.2.1 Training and test data

Topic unigram language models were trained using the data used for sentence segmentation. For training the language models, we removed stories with fewer than 300 and more than 3000 words, leaving 14,495 stories with an average length of 432

[5] The frequent word *oyun* (game) has another morphological parse, meaning "your vote", hence the appearance of the root *oy* (vote).

Table 9. *The most frequent nouns in a cluster, containing mostly soccer news articles*

Word	Freq	Meaning
gol	2562	goal
maç	2412	match
oyun	2071	game
takım	1659	team
futbol	1492	soccer
oy	1429	vote
yarı	1275	half
top	1257	ball
Galatasaray	1230	Galatasaray
Beşiktaş	1201	Beşiktaş
saha	1189	field
Fenerbahçe	1162	Fenerbahçe
dakika	1029	minute
orta	868	middle
rakip	852	opponent
lig	695	league
kale	657	goal
dk.	642	min.
pozisyon	638	position
hata	606	error
teknik	594	technical
Hakan	579	Hakan
hakem	543	referee
alan	541	space or field

words, 500 stems, or 310 nouns, excluding stop words, for a total of 376,371 distinct words, 128,125 distinct stems, or 119,475 distinct nouns.

We evaluated our system on a test set of 100 news articles, covering the period from September 12 1998 through September 14 1998, comprising 2803 sentences, 32,772 words, 38,329 stems, or 24,807 nouns, excluding stopwords. The topic switch penalty was optimized on the development set of 99 news articles from the same newspaper, between September 14 1998 and September 16 1998, comprising 3,180 sentences, 33,728 words, 39,106 stems, or 25,615 nouns, excluding stopwords.

3.2.2 Evaluation metrics

We have adopted the evaluation paradigm used by the TDT2 (Doddington 1998) program, allowing fair comparisons of various approaches both within this study and with respect to other recent work. Segmentation accuracy was measured using TDT evaluation software from NIST, which implements a variant of an evaluation metric suggested by Beeferman, Berger and Lafferty (1999).

3.2.3 Topic segmentation results

Table 3.2.3 shows the results of the Turkish topic segmenter, using word-based, stem-based, and noun-based approaches. Chance performance indicates the result,

Table 10. *Summary of error rates with different language models. A "chance" classifier that labels all potential boundaries as non-topic would achieve 0.3 weighted segmentation cost. "Random" indicates that the articles are shuffled*

Model	Development set			Test set		
	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}
Chance	1.0000	0.0000	0.3000	1.0000	0.0000	0.3000
Human Performance	0.2093	0.0176	0.0742	N/A	N/A	N/A
Word-based	0.4394	0.0658	0.1779	0.3560	0.0752	0.1594
Word-based (Random)	0.3412	0.0286	0.1224	0.3840	0.0427	0.1451
Stem-based	0.2704	0.0655	0.1270	0.2552	0.0708	0.1261
Noun-based	**0.2627**	**0.0413**	**0.1077**	**0.2487**	**0.0492**	**0.1090**

when we mark all the boundaries as non-topic boundaries, i.e. 100% misses, but no false alarms. We also tried to obtain the human performance. In this case, the overall performance of the human happened to be 7.42%, indicating that this is not a trivial task.

These results are consistent with our intuition, that we have tried to explain in the previous section. As expected, the word-based model suffered from data sparseness, and 28.61% relative improvement is achieved for the development set when we use the stems of the words. Furthermore, it is possible to obtain an additional 15.19% relative improvement using only nouns, achieving a total of 39.46% relative (7.02% absolute) improvement over our baseline word-based model. For the test set, the results are also similar, and we achieved 20.89% relative improvement when we used the stem-based approach, and our results are 31.61% relatively (5.04% absolutely) better when we used the noun-based approach.

Comparing these three modeling approaches, we observe that stem-based and noun-based models have a 38–40% lower miss probability than the word-based model in the development data. This rate is 28–30% in the test set. This enormous decrease in the miss probability is the main reason of the final improvement. We would say that, using stems, we have obtained more discriminative topic unigram language models in the clustering phase, hence we have missed fewer topic boundaries. Additionally, when we have used the noun-based models, we see that there is a 31–37% relative improvement over the stem-based models in the false alarm probabilities.

Let us analyze these results using a concrete example. Consider the following sentence from an article on soccer: *Son dakikalarda Galatasaray'ın atakları sıklaştı, Hakan attığı golle ağları sarstı.* (Literally, "In the last minutes, Galatasaray's attacks became more frequent, Hakan shook the net with the goal he scored.") Table 11 shows the individual unigram probabilities of the words in a cluster including mainly soccer news articles for both word-based and stem-based approaches. Note that, due to data sparseness, all of these words, though related with soccer have less probability when compared to stem-based and noun-based models. Furthermore,

Table 11. *The unigram probabilities of the words in the example sentence. Note that, the word* son *(last) is a stopword, hence gets 0 probability*

Word	Morphological Analysis	Word-based Probability	Stem-based Probability	Noun-based Probability
Son	`Last+Adj`	0	0	0
dakikalarda	`minute+Noun+A3pl+Pnon+Loc`	0.000337	0.004930	0.007296
Galatasaray'ın	`Galatasaray+Noun+Prop+A3sg+Pnon+Gen`	0.001433	0.005598	0.008679
atakları	`attack+Noun+A3pl+P3sg+Nom`	0.000072	0.001192	0.001600
sıklaştı	`frequent+Adj^DB+Verb+Become+Pos+Past+A3sg`	0	0.000557	0
Hakan	`Hakan+Noun+Prop+A3sg+Pnon+Nom`	0.002556	0.005422	0.004087
attığı	`score+Verb+Pos^DB+Adj+PastPart+P3sg`	0.001232	0.005458	0
golle	`goal+Noun+A3sg+Pnon+Ins`	0.000760	0.012454	0.018019
ağları	`net+Noun+A3pl+Pnon+Acc`	0.000138	0.000428	0.000595
sarstı	`shake+Verb+Pos+Past+A3sg`	0.000001	0.000127	0

the word *sıklaştı* (became frequent) received 0 probability, since its surface form is unseen in the training data, although its stem *sık* (frequent) gets some probability. Note that, we preferred not to smooth the probabilities, in order to capture the differences between the topic clusters.

3.2.4 Error analysis

When we analyze our errors, we see that errors are made when there are topically very similar news articles in a sequence, or when an article contains more than one topic, though this second case is less likely. This is why we obtained better performance on the test set than the development set for both word-based and stem-based models, although we set the topic switch penalty on the development set. When we analyzed this, we see that development set is *harder* to segment than the test set, in the sense that it includes articles with very similar consecutive topics. Note that, because of this, the miss probability of a human annotator is about 20%. When we ordered the articles randomly, this difference disappeared.

3.2.5 Results compared to topic segmentation of English

It would be useful to provide word-based segmentation error rates obtained from a recent work (Tür, Hakkani-Tür, Stolcke and Shriberg 2001) for English Broadcast News corpus. As shown in Table 12, the two test sets have comparable behavior. Stem-based and noun-based models are not available for English. It would be interesting to try these approaches for English, too.

4 Name tagging

One of the basic tasks in an information extraction system 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 names is called *name tagging*.

Table 12. *Word-based segmentation error rates for English and Turkish corpora*

Corpus	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}
Turkish	0.4394	0.0658	0.1779
English	0.4685	0.0817	0.1978

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 1995; MUC 1998). A very detailed definition of the named entity extraction task has been developed in the framework of these programs (Chinchor and Robinson 1998).

4.1 Approach

Our approach is based on *n*-gram language models embedded in hidden Markov models. We used the following four 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.

4.1.1 Lexical model

For lexical modeling, we used a simplified version of BBN's name finder (Bikel, Schwartz and Weischedel 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.

To detect the boundaries of the names, we used a fictitious boundary flag. This flag holds one the following three 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 3. Note that, although it is possible to get a sequence of "person mid organization", the use of language

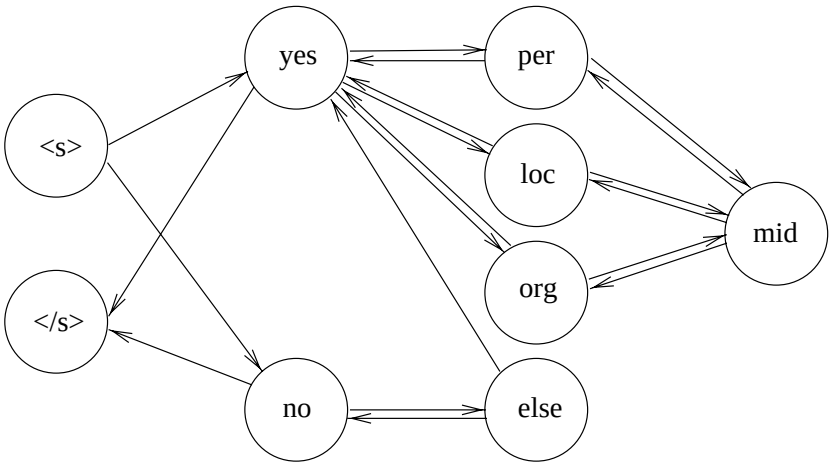

Fig. 3. The conceptual structure of the basic HMM for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, yes denotes the name boundary, no denotes that there is no name boundary, mid denotes that it is in the middle of a name, per denotes person, loc denotes location, org denotes organization, and else denotes that it does not belong to any of these categories.

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 three name types.

An example will clarify this notation. Consider the 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* Business/*organization* boundary/*yes* is/*else* boundary/*no* in boundary/*yes* Ankara/*location* boundary/*yes* </s>"

where <s> denotes the start of sentence, and </s> denotes the end of sentence.

This implies that, name tagging task does not only require tagging each word with one of the four possible tags (person, location, organization, and else), but also detecting the boundaries. In fact, using this boundary flag also 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 Table 13, the boundary flag helps us to find the correct tagging, since the trigram "Ankara/*organization* boundary/*mid* Üniversitesi/*organization*" is about

Table 13. *The effect of the boundary flag on the performance of the tagger*

Output sequence	Probability
Ankara/*organization* boundary/*mid* Üniversitesi/*organization*	0.015029
Ankara/*location* boundary/*yes* Üniversitesi/*organization*	0.000004

Table 14. *The use of the contextual model for unknown words*

Output Sequence	Probability
Dr./*else* boundary/*yes* unk/*person*	0.990119
Dr./*else* boundary/*yes* unk/*location*	0.000690
Dr./*else* boundary/*yes* unk/*organization*	0.000880
Dr./*else* boundary/*else* unk/*else*	0.002688

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*". This is why marking the whole phrase as a location is more probable than separating them.

4.1.2 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, Tür, Stolcke and Shriberg 1999). For example, the word after the abbreviation "Dr." is generally a person, The word "University" is often a part of an organization.

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 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 14 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 clues:

1. Previous token in the same name, e.g. First names of the persons in a context like "Gökhan Tür", assuming that first names are a smaller set than the surnames,
2. Previous token outside of the name, e.g. "Mr.", "Dr.", in a context like "Dr. Tür",
3. Next token in the same name, e.g. "Üniversitesi", in a context like "Manitoba Üniversitesi" (Manitoba University),
4. Next token outside of the name, e.g. "kentinde", in a context like "İstanbul kentinde" (in the city of Istanbul).

These cues can be considered as the help of prepositions in English. Since, Turkish is an agglutinative language, there are no prepositions, but corresponding suffixes are attached to 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.

4.1.3 Morphological model

We built the morphological model, similar to the one developed for sentence segmentation. Additionally, we inserted 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. Besides this, 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.

Table 15. *The use of the morphological model*

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

Let us demonstrate these expectations using a concrete example. Similar to Tables 13 and 14, Table 15 gives the probabilities for the named entity:

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

where, both "Süleyman" and "Demirel" are analyzed as:

"Noun+Prop+A3sg+Pnon+Nom+CAP".[6]

4.1.4 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 seven tokens. We built it by extracting the lexical words in our training data, and leaving only these tags.

The idea of developing a tag model was suggested by the result of the analysis of 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>
```

On the other hand, the tag models favors for the correct tagging as seen in Table 16.

In other words, the function of this model is to limit the improbable 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.

[6] See Appendix A for the definition of features in this morphological parse.

Table 16. *The use of the tag model*

Output sequence	Probability
person mid person	0.999870
person yes person	0.006076

4.1.5 Model combination

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

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

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

Combining the 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 the 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 the 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 probability

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

$P(w_i/t_i) = P(t_i|w_i)$, since w_i is given. Hence, we can rewrite the above formula 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 probability:

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

where MM denotes morphological model, $M(w)$ denotes the morphological analyses

of the word w. Following the above notation we can say that this posterior probability is equal to:

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

Then, we can simply interpolate these posterior probabilities with some weight λ. A top level representation of this interpolation can be written 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 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 to determine the transaction probabilities. One problem with this operation is converting posterior probabilities, $P(T|W)$, to likelihoods, $P(W|T)$. This conversion is possible using Bayes' rule:

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

Since we use 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:

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

where CM denotes the contextual model using contexts of the words, $C(W)$, TM denotes the 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 4 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 posterior probabilities from the lexical model, convert to likelihoods, and decode using the tag model as transition probabilities.

4.2 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,

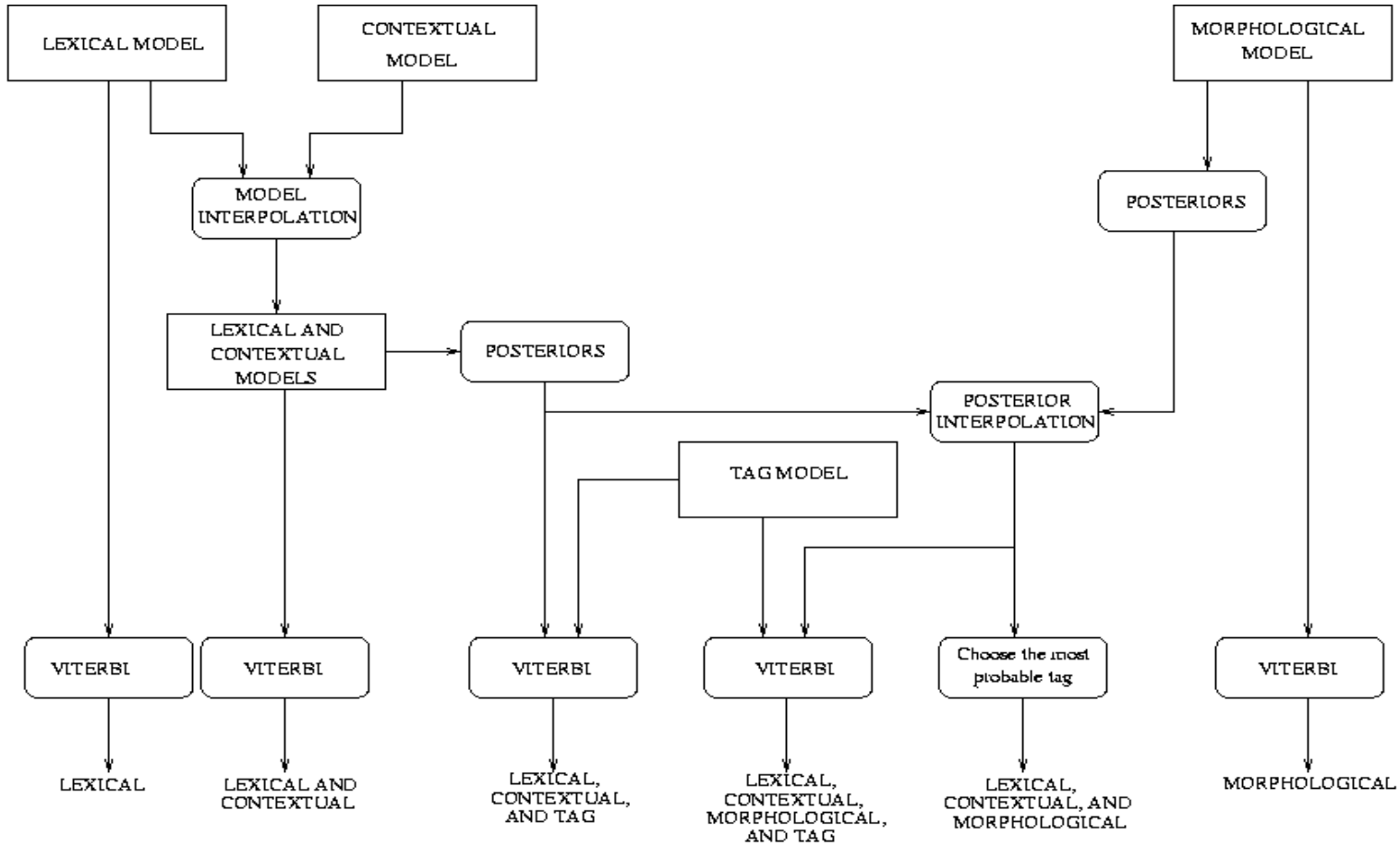

Fig. 4. Combining lexical, contextual, morphological, and tag models for tagging Turkish text.

and their combinations, we also present results for tagging English, using the same models and evaluation metrics.

4.2.1 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 2197 names.

4.2.2 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.

The overall accuracy result, *F-Measure*, is computed by the uniformly weighted harmonic mean of precision and recall:

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

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 compared to the number of answers delivered.

4.2.3 Name tagging results

Table 4.2.3 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 four types of combinations shown in the table, although it is possible to combine these information sources in eleven different ways. In all of the combinations, we did not separate the lexical model from the contextual model, because the lexical model alone is relatively very weak in tagging. So we are left with only four types of combinations.

We are very pleased to see that, the lexical model alone performed in the 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 for this problem is the unknown words. This problem is solved by the contextual model, and the performance, using the "Text" metric is increased to 86%. It is also interesting to see that the morphological model alone has performed 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% more. Instead of the morphological model, when we have incorporated the tag model, we have gained about 2% more. These

Table 17. *Accuracy of the name tagging task using lexical, contextual, morphological, and tag models*

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 18. *Detailed name tagging results*

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

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. Using the "Text" metric, the performance is increased more than 3%, whereas using the "Type" metric, this number is about 0.5% in either case this model was used. Similarly, the morphological model increases the F-Measure by 0.8% in either case it was used. When we compare the final F-Measure with our baseline lexical performance, we see an improvement of 5.55%.

4.2.4 Error analysis

Table 4.2.4 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 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>
```

Table 19. *Comparison of the Turkish and English name tagging results using only lexical and contextual models*

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

was tagged as:

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

which results in two different names, neither of which were tagged completely.

4.2.5 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 4.2.5 presents the performance of our algorithm applied to both English and Turkish.

5 Conclusions

We have presented statistical solutions to various information extraction tasks for Turkish. Statistical methods have been largely ignored for processing Turkish. Mainly due to the agglutinative nature of Turkish words and the structure of Turkish sentences, the construction of a language model for Turkish can not be directly adapted from English. It is necessary to incorporate some other techniques. This work is a preliminary step in the application of corpus-based statistical methods to Turkish text processing. Future work includes using more sophisticated methods, like maximum entropy models. For sentence segmentation, we have modeled the final inflectional groups of the words and combined it with the lexical model, and decreased the error rate to 4.34%, which is 21% better than the result obtained using only the surface forms of the words. For topic segmentation, stems of the words (especially nouns) have been found to be more effective than using the surface forms of the words and we have achieved 10.90% segmentation error rate on our test set according to the weighted TDT-2 segmentation cost metric. This is 32% better than the word-based baseline model. For name tagging, we used four different information sources to model names. 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 some 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%, according to the MUC evaluation criteria. According to the

McNemar's test, all our improvements are statistically significant, with a p value of less than 0.0001. 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 significantly improves these basic information extraction tasks for Turkish.

A Turkish morphological features

Feature	Definition
^DB	Derivation boundary
A3sg	Third person singular agreement
A3pl	Third person plural agreement
Abl	Ablative case
Acc	Accusative case
Adj	Adjective
Agt	Agent
Become	Become verb
Caus	Causative verb
Dat	Dative case
Det	Determiner
FitFor	FitFor
Gen	Genitive case
Ins	Instrumantive case
Loc	Locative case
Nom	Nominative case
Noun	Noun
P1sg	First person singular possessive agreement
P2sg	Second person singular possessive agreement
P3sg	Third person singular possessive agreement
P3pl	Third person plural possessive agreement
PastPart	Derived past participle
Pnon	No possessive agreement
Pos	Positive polarity
Prop	Proper name
Verb	Verb
With	With
Without	Without
Zero	Zero derivation with no overt derivation

Acknowledgements

This work was begun while the first two authors were visiting Speech Technology and Research Laboratory, SRI International, as international fellows, 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.

References

Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y. (1998) Topic detection and tracking pilot study: final report. *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194–218. Lansdowne, VA.

Beeferman, D., Berger, A. and Lafferty, J. (1999) Statistical models for text segmentation. *Machine Learning*, **34**(1–3):177–210.

Bikel, D. M., Schwartz, R. and Weischedel, R. M. (1999) An algorithm that learns what's in a name. *Machine Learning*, **34**:211–231.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L. and Roossin, P. S. (1990) A Statistical Approach to Machine Translation. *Computational Linguistics* 16(2):79–85.

Chinchor, N. and Robinson, P. (1998) MUC-7 Named Entity Task Definition (version 3.5). *Proceedings of the MUC-7*.

Church, K. W. (1988) A stochastic parts program and noun phrase parser for unrestricted text. *Second Conference on Applied Natural Language Processing*, pp. 136–143. Austin, Texas.

Doddington, G. (1998) The topic detection and tracking phase 2 (TDT2) evaluation plan. *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, pp. 223–229. Lansdowne, VA.

Grishman, R. and Sundheim, B. (1996) Message Understanding Conference-6: A Brief History. *Proceedings 16th International Conference on Computational Linguistics*, Copenhagen, Denmark.

Grishman, R. (1998) Information Extraction and Speech Recognition. *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA.

Hakkani-Tür, D. and Oflazer, K. (2000) Statistical morphological disambiguation for agglutinative languages. *Proceedings 18th International Conference on Computational Linguistics*, Saarbrucken, Germany.

Hakkani-Tür, D., Tür, G., Stolcke, A. and Shriberg, E. (1999) Combining words and prosody for information extraction from speech. *Proceedings 6th European Conference on Speech Communication and Technology*, vol. 5, pp. 1991–1994. Budapest.

Hakkani-Tür, D. Z. (2000) Statistical Language Modeling for Turkish. PhD Dissertation, Department of Computer Engineering, Bilkent University, Ankara, Turkey.

Hankamer, J. (1989) Lexical representation and process. In: Marslen-Wilson, W., editor, *Morphological Parsing and the Lexicon*. MIT Press.

Kozima, H. (1993) Text segmentation based on similarity between words. *Proceedings 31st Annual Meeting of the Association for Computational Linguistics*, pp. 286–288. Columbus, Ohio.

MUC-6 *Proceedings of the MUC-6*.

MUC-7 *Proceedings of the MUC-7*.

Oflazer, K. (1993) Two-level description of Turkish morphology. *Literary and Linguistic Computing* **8**(3).

Oflazer, K. (1999) Dependency parsing with an extended finite state approach. *Proceedings 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland, College Park, MD.

Palmer, D. D. and Hearst, M. A. (1997) Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics* **23**(2):241–267.

Reynar, J. C. and Ratnaparkhi, A. (1997) A maximum entropy approach to identifying sentence boundaries. *Proceedings 5th ACL Conference on Applied Natural Language Processing*, Washington, DC.

Shriberg, E., Stolcke, A., Hakkani-Tür, D. and Tür, G. (2000) Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication* **32**(1-2).

Stolcke, A., Shriberg, E., Bates, R., Ostendorf, M., Hakkani, D., Plauché, M., Tür, G. and Lu, Y. (1998) Automatic detection of sentence boundaries and disfluencies based on recognized words. In: R. H. Mannell and J. Robert-Ribes, editor, *Proceedings International Conference on Spoken Language Processing*, vol. 5, pp. 2247–2250. Sydney.

Stolcke, A., Shriberg, E., Hakkani-Tür, D. Tür, G., Rivlin, Z. and Sönmez, K. (1999) Combining words and speech prosody for automatic topic segmentation. *Proceedings DARPA Broadcast News Workshop*, pp. 61–64. Herndon, VA.

Stolcke, A. (1999) SRILM – the SRI language modeling toolkit. http://www.speech.sri.com/projects/srilm/.

Tür, G., Hakkani-Tür, D., Stolcke, A. and Shriberg, E. (2001) Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics* **27**(1).

Tür, G. (2000) *A Statistical Information Extraction System for Turkish.* Ph.D. Dissertation, Department of Computer Engineering, Bilkent University, Ankara, Turkey.

Mulbregt, P. van, Carp, I., Gillick, L., Lowe, S. and Yamron, J. (1998) Text segmentation and topic tracking on broadcast news via a hidden Markov model approach. In: R. H. Mannell and J. Robert-Ribes, editors, *Proceedings International Conference on Spoken Language Processing*, vol. 6, pp. 2519–2522. Sydney.

Wayne, C. L. (1998) Topic Detection and Tracking (TDT) Overview and Perspective. *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA.

Yamron, J. P., Carp, I., Gillick, L., Lowe, S. and van Mulbregt, P. (1998) A Hidden Markov Model approach to text segmentation and event tracking. *Proceedings IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 333–336. Seattle, WA.