Exploring the Tag Space for Morphosyntactic Tagging without Morphological Analysis

Utku Şirin
Teknoloji Yazılımevi, Ltd.
Ankara, Turkey
utkusirin@gmail.com

Selçuk Köprü
Teknoloji Yazılımevi, Ltd.
Ankara, Turkey
selcuk.kopru@tyazilimevi.com

Cem Bozşahin
Informatics Institute, METU
Ankara, Turkey
bozsahin@metu.edu.tr

Abstract

This paper describes our work on morphosyntactic tagging and our exploration of different tagsets for Turkish. Our tagging approach is based on an adaptable Hidden Markov Model. It uses suffix and stem probabilities for Turkish for out-of-vocabulary words. Turkish is a morphologically rich language and it tends to require large tagsets. Our simple method and its efficient stochastic tagger has given comparable results without the help of a morphological analyzer. We have also conducted several experiments on reducing the tagset size of Turkish. We make use of the Frequent Pattern tree (FP-tree) data structure for this task, which revealed different design choices. We have achieved an accuracy of 94.20% without any reduction on the richest tagset (5,917 tags), and 98.42% on basic 20 POS tags. To our knowledge, 94.20% accuracy is the highest achieved result on full morphosyntactic tagging of Turkish without a morphological analyzer. This is a first attempt at exploring the possibility of tagset performance depending on the task in hand.

1 Introduction

Morphosyntactic tagging can be defined as the problem of finding a tag for a word to represent both morphological and syntactic properties of the word. Although part-of-speech (POS) tagging is also a kind of morphosyntactic tagging, in this work we use the latter term because the tags we are concerned with include as much morpheme markers in them as possible, not only basic parts of speech.

For highly inflected and agglutinating languages in particular, this task is very important due to richness of the tags. Morphological structure of these languages allows us to "agglutinate" the words, and have many number of morphemes within a single word. Since each morpheme means a new piece of information, this process results in informational accumulation on words, and makes understanding and modeling such languages directly relate to understanding and modeling the words and their morphosyntactic properties.

One important result of agglutination and inflection is the number of possible tags that can be assigned to a word. For Turkish in particular, there is no upper bound on suffixation. There may be infinitely many possible tags. For example, the morpheme sequence "-de-ki" (-LOC-ki) can repeat any number of times as long as it can find the right stem (Hankamer, 1989; Göksel and Kerslake, 2005): *Ev-de-ki* (house-LOC-ki) 'the one in the house', and *ev-de-ki-ler-de-ki* (house-LOC-ki-PLU-LOC-ki), literally meaning 'the one in the ones of the house' e.g. referring to the picture belonging to the family in the house.

The recursive nature of the process creates many recurrent tags. While the tagset size for morphologically simpler languages such as English is less than one hundred, or around hundred with the full tagset of for example Penn Treebank, the same number for highly inflected agglutinating languages easily exceeds—without recursion—one thousand (Hakkani-Tür et al., 2002; Varadi and Oravecz, 1999; Hajic, 2000; Dzeroski et al., 2000).

Large tagsets complicate the tagging problem for

morphologically rich languages. It is one of the reasons why most studies on tagging attempt morphological disambiguation rather than morphosyntactic tagging for these languages, especially for Turkish (Hakkani-Tür et al., 2002; Yüret and Türe, 2006; Sak et al., 2007). The tacit assumption seems to be that an effective solution to the tagging problem is almost impossible due to large tagset sizes. Solving the morphological disambiguation problem, on the other hand, has a head start because it uses an output of a morphological analyzer to choose the correct morphological parse out of its output (Elworthy, 1995).

Cross-linguistically, we would expect the same asymptotic behavior in all languages about the average morphosyntactic information carried by a word in syntactic contexts. We see a convergence in this regard as the datasets become larger and purposes diversify: Penn trebank's full tagset (with dashed and slashed tags) measures in hundreds when used for languages other than English. Czech treebank has 970 tags (Hajic, 2000), Slovene 1,021 (Dzeroski et al., 2000), Hungarian 571 (Oravecz and Dienes, 2002), Romanian 611 (Tufiş and Mason, 1998), and Turkish treebank 1,350 (Oflazer et al., 2003). When tagging aims to aid parsing, even a morphologically simpler language needs tags in the order of hundreds or thousands: Clark and Curran (2007) use 425 "supertags" for English that occur more than 10 times (Hockenmaier and Steedman (2007) use 1,286 lexical tags for wide-coverage parsing of English). Rich morphology might show complex word structure and may result in large tagset sizes, but total information on a large corpus is expected to be about the same across languages. The problem, we think, is not the richness of morphology but the dispersion of information that should be collected by an effective and comprehensive tagset. By designing such a tagset, the tagging problem can be handled in a simple and fast way without the help of a morphological analyzer. We offer in this paper a way of choosing among the tagsets for varying and sometimes conflicting purposes.

Yüret and Türe (2006) have tried as part of their experiments tagging without morphological analysis. Their main focus, however, is morphological disambiguation, not tagging. They do not explore tagset alternatives. They report a tagging accuracy of 91.23% which is comparable to our best result 94.20%. One of the main contributions of the current paper is to fill the gap of tagset design and morphosyntactic tagging of Turkish without using a morphological analyzer, and without loss of performance.

The rest of the paper is organized as follows: Section 2 provides a background on tagging in agglutinating languages. Section 3 describes the tagset design in detail. Section 4 reports the experiments on tagset choice and their results.

2 Background

Most of the studies on disambiguation use their system as a tagger as well. However, they either use the output of a morphological analyzer, or keep their tagset only at the level of basic part of speech tags.

Oflazer and İlker Kuruöz (1994), Oflazer and Tür (1997) are rule-based approaches to morphological disambiguation and tagging of Turkish. Hakkani-Tür et al. (2002) build a statistical system for morphological disambiguation based on inflectional groups and root probabilities, for obtaining the most probable tag sequence from a candidate set of morphological parses. They also use their system as a POS tagger by singling out the tag of the word-final inflectional group, which gives a 96.07% accuracy for that task. Yüret and Türe (2006) introduce a new method based on decision lists for learning the disambiguation rules from context. They trained a single decision list using the full tags as the target classification for comparison, and obtained a 96.03% performance for disambiguation. Sak et al. (2007) implement the perceptron-based algorithm of Collins (2002). They first obtain the n-best candidates of alternative morphological parses of a sentence from the trigram model of Hakkani-Tür et al. (2002), then implement a set of features to rerank the possible candidates. In another study (Sak et al., 2011) they do a Viterbi decoding of the best path in the network of ambiguous morphological parses. They have also used their system as a POS tagger, and reported 98.27% accuracy in (Sak et al., 2007), and 98.6% in (Sak et al., 2011). There is also a suffix-based POS tagger for Turkish (Dinçer et al., 2008). They fix the number of characters of a word as *length*$=1...7$, and apply Viterbi algorithm over

simple HMMs. Their 90.2% success rate is best with a 5-gram.

There has been a good amount of work on tagset design for other morphologically rich languages such as Slovene, Czech, Hungarian and Romanian. They have used full morphosyntactic descriptions and applied some techniques for reducing the tagset, such as removing case, gender and number markers (Dzeroski et al., 2000; Varadi and Oravecz, 1999), or attempted to reduce the tagset which can be deterministically mapped to the full tagset (Tufiş and Mason, 1998). Dzeroski et al. (2000) obtained 89.22% accuracy for Slovene with the TnT tagger of Brants (2000), an HMM-based trigram tagger, with a tagset size of 1,021 including the full morphological parse results as morphosyntactic tags of words. They report 96.59% accuracy for finding basic POS tags with a tagset of size 12. Oravecz and Dienes (2002) also used the TnT tagger, for Hungarian, and measured 92.88% accuracy for full morphosyntactic tagging with a tagset of size 571. Tufiş and Mason (1998) have built a reduced tagset, called C-tagset, with a size of 89 tags from a complete tagset size of 611 for Romanian. They call the method "Tiered Tagging", which first finds a tag from the C-tagset for a word, then maps the tag to its complete form with almost 98% success. They report 97.82% accuracy with their tiered-tagging approach. Hajic (2000) built a universal exponential feature-based model for comparing tagging on several languages. They measure 92.96% success with a tagset of 970 for Czech.

3 Tagging and Tagset Design

In our study we focus on a bigram HMM model which works without a morphological analyzer. We start with the morphosyntactic tags proposed for Turkish, which, over 5 million words, gives a full tagset size of 5,917. For our semi-automatically disambiguated dataset of 1 million words, the tagset is its subset, 4,385. We obtained 94.20% accuracy of morphosyntactic tagging with 5,917 tags. This result is higher than the only previous study on tagging of Turkish without a morphological analyzer (Yüret and Türe, 2006). Clearly, having that many tags jeopardizes the balance we seek by providing too much information than can be useful, and using a small set of 20 provides too little to be of general use. (We obtained 98.42% accuracy with 20 tags.) Very high accuracies at these extremes could therefore be misleading unless put in context. We considered several tagset choices and their accuracies for Turkish. We introduce a novel use of the frequent-pattern tree for tagset exploration, to automatically build a tagset based on morpheme frequencies.

3.1 Morphosyntactic Tagging of Turkish

POS tagging is a classification problem, and the HMM is very successful in such problems. Our work is inspired by systems which employ the HMM approach to POS tagging in different languages, such as Brants (2000). We use the tagger of Köprü (2011), which implements the well-known model of HMM:

$$\hat{t}_1^n = \arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \qquad (1)$$

In Equation (1), the sequence of words in the observed sentence is represented as $w_1 \dots w_n$. In order to calculate the estimate of the tag sequence $\hat{t}_1^n$, tag transition probabilities and word likelihoods are used. The tag transition probability $P(t_i|t_{i-1})$ is estimated with a bigram Language Model (LM) that is constructed from a tag transition corpus. The word likelihood $P(w_i|t_i)$ is computed from counts in the training data. The HMM is represented as a weighted finite state machine (FSM) with hidden states. In tagging, the HMM states correspond to the tags and the output symbols represent the words. Viterbi algorithm is used to determine the most likely tag sequence.

3.2 Tagset Design for Turkish

Much of the information about a language is embedded in words for highly inflected and agglutinating languages. Tagsets thus directly relate to revealing the grammatical information on words. From this perspective, tagset design means building a set of tags which include as much information about the structure in words as possible while maintaining efficiency in tagging.

We carried out two sets of tagset design experiments. The first one is based on unification of some

linguistic aspects. The second one is based on analyzing the frequencies of markers via a frequency tree.

3.2.1 Unification of Linguistic Markers

The first set of experiments with the full tagset is motivated by linguistic concerns. For example, we know that agreement morphology is abundant in Turkish, not only for subject-verb agreement in finite clauses, but also in noun-noun agreement in genitive constructions and embedded subject-embedded verb agreement in subordination, which are very frequently used constructions. This is a syntactic glue for semantics in Turkish grammar. Naturally, agreement tends to diffuse information *per word* because the dependency is not within words but phrases.[1] We also know that as an agglutinating language, the word-final POS tag must be crucial for syntax. Indeed, some wide-coverage Turkish parsers make use of only that information (Oflazer, 2003; Eryiğit et al., 2008). Our results seem to provide an empirical justification for these generalizations.

This part of our experiments is inspired by previous studies on tagset design for languages mentioned in §2. We extend the experiments to unify the most frequent inflectional groups, namely, case, person, tense and possessive markers and their combinations. Our main experiments are listed below.

1. No conflation of markers
2. Unifying ABL, ACC, DAT, LOC, GEN and INST case markers as CASE
3. Unifying 1ST, 2ND, 3RD person SINGULAR and PLURAL POSS markers as POSS
4. Unifying PAST, NARRATIVE, FUTURE, AORIST and PROGRESSIVE tense markers as TENSE
5. Unifiying 1ST, 2ND, 3RD person SINGULAR and PLURAL personal markers as PERSON
6. Unifying the markers specified in items 1 and 2
7. Unifying the markers specified in items 3 and 4
8. Unifying all markers
9. Unifying the markers specified in items 1 to 4, and unifying the combined tags POSS and CASE markers as POSS-CASE, and PERSON and TENSE as TENSE-PERSON.

[1] In one experiment, Clark and Curran (2007), the dictionary cutoff of 20 worked well for English. It certainly was too high for Turkish because even some function words could not make the cut (Çakıcı, 2008). This we believe corroborates further that we need experiments with tagsets rather than take them as given or fixed for a parser.

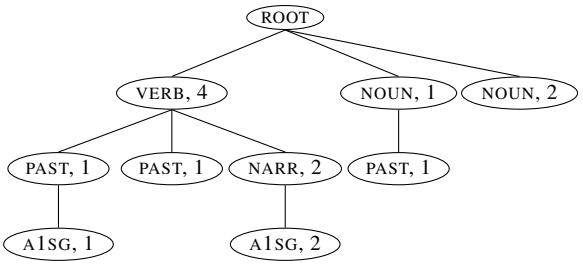

Figure 1: Sample tags and the corresponding FP-tree.

The results we report in §4 refer to these experiments by label.

3.2.2 FP-tree

The Frequent Pattern-tree (Han et al., 2004) is a tree of labels (morpheme tags in our case), where each node keeps a label and its frequency. We built ours over the tags of our semi-automatically disambiguated 950,000-word corpus and automatically disambiguated 4 million-word corpus. For each marker of a morphological parse, we add a node to the tree, and increase its frequency in the order that the marker is situated in the parse result. Hence, each morphological tag sequence corresponds to a path in the tree, and each frequency denotes the number of occurrences of a path in the corpus. For example, for the tags below, we derive the frequency tree in Figure 1. Each node frequency is a total of its children's frequencies.

V-PAST-A1SG V-PAST V-NARR-A1SG N
V-NARR-A1SG N-PAST N

The main benefit of this tree is that the information related to tags is represented in a structural way. We can do experiments on the tagset easily just by aiming to maximize informational complexity of the tree by grouping its nodes, to understand the behavior of a language in a data-driven way.

4 Experiments

For training, we first assume the morphosyntactic tag of a word as its full morphological parse. Such results are available thanks to morphological analyzers and disambiguators. The tagset size from these studies is 4,385. This tagset is derived from the 950,000-word semi-automatically disambiguated data, and includes all possible tags that may apply to a word (see §4.1). Our method avoids analysis by using these "morphological supertags", which are sequences of morpheme tags associated with one word, as part of the lookup target pair (raw data × its supertag in training) for a test word in morphosyntactic tagging.

For example, the word *dünyasının* is morphologically *dünya-sı-nın* (world-N-POSS3S-GEN3S) 'of his/her world' in the gold standard, where the sequence beginning with the first dash is its morphological supertag (the remainder is considered to be the stem, in this case *dünya*). It not only includes the edge (or final) features but the entire morphological structure. This is provided by the analyzers and disambiguators mentioned in §2. When the training set is large enough there is information about various wordforms of *dünya* (hence more supertags) to feed the LM (§3.1).[2] If a word in the test set matches *dünyasının*, then its tag is already established. If not, LM provides the most likely tag given the training data.

Although the base tagset for the supertags gives us a comparable tagging accuracy, we tried to experiment with the tagset to understand how much information we lose and how much success we gain if the full set is reduced.

We use the word error rate (WER) metric while reporting the accuracy results in the experiments.

4.1 Data

We make use of two different corpora in our experiments. For the tagset exploration experiments, we use the semi-automatically disambiguated corpus of Yüret and Türe (2006) which has around 1 million tokens. The second corpus, which is used only for bootstrapping purposes, contains 4 million tokens which we morphologically parsed with Oflazer's (1994) morphological parser, and disambiguated it using Sak et al. (2007) morphological disambiguator. We use 96% of the 1-million corpus for training, 2% for development and 2% for testing. Sak et al. (2007) use the same dataset as follows: 750,000 (79%) for training, 40,000 (4.2%) for development, 40,000 for testing). We use the development data for optimizing the suffix/stem weights. These parameters are used for calculating the tag probability of words that are not fully matched.

[2]The model is adaptable to morphological typology. For example, in Köprü (2011), the stem-and-affix model of LM becomes a prefix-template-suffix model for Arabic.

No	Unification Type	Tagset Size	Accuracy
1	Full tagset	4,385	91.16%
2	CASE	3,369	91.40%
3	POSS	3,603	91.93%
4	TENSE	3,631	91.24%
5	PERSON	3,941	91.18%
6	CASE, POSS	2,829	92.70%
7	TENSE, PERSON	3,301	91.29%
8	CASE, POSS, TENSE, PERSON	1,834	93.00%
9	CASE, POSS, TENSE, PERSON, POSS-CASE, TENSE-PRES	1,355	94.02%

Table 1: Experiments based on the unification of linguistic markers

4.2 Linguistic Marker Experiments

We have tested our data by unifying some frequent inflectional groups and their combinations. Table 1 lists the results of the linguistic experiments.

In these results, the POSS marker presents itself as the most efficient unified inflectional group (cf. first five rows of Table 1). The potential explanation for that was provided before (§3.2.1), which has to do with pervasiveness of Turkish agreement morphology. It mainly coincides with possessive marking.

Experiments 8 and 9 score the best accuracy results. This seems natural given the fact that these experiments lump almost all Turkish inflection into one group (except voice and negation). Experiments 6–7 do the same in smaller groups. Experiment 6 cross-cuts nominal and verbal inflection because what is glossed as POSS applies to both paradigms. We think its result expectedly stands out among the

first seven because of that reason. Notice also the resulting tagset size in total grouping of inflections, 1,355, which seems to be close to cross-linguistic findings reported earlier for this level of accuracy. This result is quite striking when considered in light of frequency patterns. From FP-tree level tests (experiments 20-27 in §4.3 and Table 4), we see that at level 3 the accuracy drops just below 92% with only 998 tags, compared to same level of performance in Experiment 6 above with 2,829 tags.

4.3 FP-tree Experiments

We also conducted experiments on the grouping of the frequency data structure without a concern for linguistic aspects of the markers. The grouping of the FP-tree is based on averages. $Freq$ represents the frequency of a node, $avg_{tree}(Freq)$ represents the average frequency of the tree, and $avg_{1,000}(Freq)$ represents the average frequency based on the most frequent 1,000 nodes. In each experiment, we set a different threshold value based on the average frequency, and prune the nodes in the tree having a frequency less than the threshold. These experiments are listed below:

10. No pruning
11. Prune nodes if $Freq < avg_{tree}(Freq)$
12. Prune nodes if $Freq < avg_{tree}(Freq)/2$
13. Prune nodes if $Freq < avg_{tree}(Freq)/4$
14. Prune nodes if $Freq < 2 * avg_{tree}(Freq)$
15. Prune nodes if $Freq < avg_{1,000}(Freq)$

Experiments 11–14 prune the leaves of the tree according to the $avg_{tree}(Freq)$ value to discard infrequent tags. The aim is to build a compact tree with a reduced number of tags and to observe the performance change under different tagsets. The average frequency in the tree is calculated as 138 for our training data. In Experiment 11, all nodes with a frequency below 138 are pruned. In Experiment 15, the average frequency of the most frequent 1,000 tags are used to prune the tree. The accuracy results for these experiments are presented in Table 2. Experiment 15 stands out. Its tagset is probably too weak to serve a parser, but it can perform well for opinion mining or sentiment analysis where lighter annotation suffices, and frequencies matter.

No	Pruning Type	Tagset Size	Accuracy
10	No pruning	4,385	91.16%
11	$Freq < avg_{tree}(Freq)$	725	91.95%
12	$Freq < avg_{tree}(Freq)/2$	917	91.65%
13	$Freq < avg_{tree}(Freq)/4$	1,231	91.41%
14	$Freq < 2 * avg_{tree}(Freq)$	615	92.67%
15	$Freq < avg_{1,000}(Freq)$	385	93.83%

Table 2: Experiments based on pruning the FP-tree according to the average frequency.

In another set of experiments (16 – 19, reported in Table 3), only the most-frequent N tags are kept and less frequent tags are mapped to the frequent ones. The aim in these experiments is to observe the relation between the number of tags and accuracy in a quantitative manner. We also measured the coverage of the most frequent tags and found that the most frequent 1,000 tags cover 99.2% of the entire corpus. In experiment 17, we mapped all less frequent tags to one of the most frequent 1,000 tags and checked the improvement in the accuracy. The mapping of the less frequent tags to the most-frequent tags is performed in the FP-tree by collapsing the child nodes to upper levels. Table 3 lists the results for this set of experiments. Note that experiments 15 and 17 are different because in Experiment 15 we remove all nodes with frequency below the average of most frequent 1,000 tags. Therefore, the tagset generated in Experiment 15 is covered by the tagset generated in Experiment 17. Based on their accuracy, among them we would choose the smaller tagset (385) to obtain a potentially more robust annotation.

The FP-tree constructed from the training corpus has 8 levels, i.e. for any word in the training corpus, the maximum number of morphemes attached to the stem is 7. In the experiments 20 to 27, the FP-tree is pruned according to its levels. The aim in these experiments is to observe the effect of the supertag length in morphosyntactic tagging. For example, in Experiment 20, only the POS of the stem is taken and all remaining morphemes are discarded

No	Pruning Type	Tagset Size	Accuracy
16	Frequent 800 tags	800	91.66%
17	Frequent 1,000 tags	1,000	91.66%
18	Frequent 1,200 tags	1,200	91.56%
19	Frequent 2,000 tags	2,000	91.49%

Table 3: Experiments based on pruning the FP-tree according to the most frequent N tags.

No	Pruning Type	Tagset Size	Accuracy
20	Up to level 1 (POS Tags)	20	97.21%
21	Up to level 2	171	93.84%
22	Up to level 3	998	91.56%
23	Up to level 4	2,621	91.53%
24	Up to level 5	3,886	91.22%
25	Up to level 6	4,290	91.18%
26	Up to level 7	4,376	91.16%
27	Up to level 8	4,385	91.16%

Table 4: Experiments based on pruning the FP-tree according to its levels.

because level 1 in the FP-tree contains the basic POS tags only. Similarly, in Experiment 21, the stem and the first morpheme are taken and the remaining morphemes are discarded. Experiments for all levels are performed and the relation between morpheme length and accuracy is observed.

Pruning the FP-tree by node levels did not perform well as can be seen from the lower accuracy rates in Table 4. For example, in Experiment 21, the first 2 levels of the FP-tree are used and the tagset size is only 171, however, the accuracy is 93.84 which is a poor performance compared to Experiment 9 with 1,355 tags and 94.02% accuracy.

In the last set of pruning experiments, we explore the relation between morpheme position and morphosyntactic tagging. The tagsets in these experiments are generated by pruning the mid-level nodes in the FP-tree while keeping the leaf-level nodes. The results are listed in Table 5. In Experiment 28, we observed the most robust tagging performance when the last morpheme in the tag is removed. A tagging accuracy of 94.40% was achieved with 1,440 tags, which is about the same tagset size

No	Pruning Type	Tagset Size	Accuracy
28	Last morpheme	1,440	94.40%
29	Last agreement at word final position	3,248	91.49%
30	All agreement at word final position	2,804	91.79%
31	Last 2 morphemes	498	95.69%
32	Last 3 morphemes	183	96.52%
33	Root & last morpheme	273	94.00%
34	Root & last 2 morph.	1,418	91.83%
35	Root & last 3 morph.	2,948	91.31%
36	Root & last 4 morph.	3,977	91.22%
37	Root & last 5 morph.	4,309	91.19%
38	Root & last 6 morph.	4,379	91.17%
39	Root & last 7 morph.	4,385	91.16%

Table 5: Exploration of the effect of last morphemes in the tag.

in other languages for that level of accuracy. This result seems to also coincide with the choice of Turkish wide-coverage parsers noted earlier, that the last tag is the most relevant to syntax and perhaps least dependent on the word itself. Its performance can be appreciated better if this result is compared with Experiment 18 (1,200 tags and 91.56% accuracy) or with Experiment 21 (171 tags and 93.84% accuracy). The closest accuracy to Experiment 28 at this tagset size is achieved in Experiment 9 (1,355 tags and 94.02% accuracy). There seems to be a balancing act here: If we remove all agreement information as we did in Experiments 29–30, accuracy drops and the tagset at least doubles in size. It seems to suggest that we must keep agreement information for parsing, rather than live with tagset sizes around 3,248 and gain no significant advantage in other tasks such as sentiment analysis etc.

Experiments 33-39 need explaining. For example, Experiment 33 keeps the POS tag of the stem and the last morpheme of any word in training, e.g. N-GEN3S for *dünya-sı-nın*, which is in fact *dünya-* -N-POSS3S-GEN3S ('of his/her world'). Given its relatively good success with a small tagset of 273, it can serve well in cases where lexical semantics may be needed, as in polarity items.

No	Explanation	Tagset Size	Accuracy
40	Baseline full tag set	4,385	91.16%
41	Baseline POS tag set	20	97.21%
42	Remove singletons	3,182	91.24%
43	Remove doubletons	2,665	91.24%
44	Data population (full tagset)	5,917	94.20%
45	Data population (POS tagset)	20	98.42%

Table 6: Baseline optimization experiments.

4.4 Baselines

Our goal in the last set of experiments is to improve the baseline system performance. Table 6 summarizes the results. Experiment 40 is same as Experiment 1, likewise Experiment 41 and 20, which are included for checking. In experiments 42 and 43, we removed the single occurrences and double occurrences of the tags, which did not improve the results very much compared to the baseline. From this result we can conclude that the training data does not contain too much noise.

All the experiments reported so far use the 1-million corpus for training. Experiment 44 repeats Experiment 40 with the addition of 4 million more words, which were morphologically analyzed and disambiguated as before. We gain 3% more in accuracy with 1,532 more tags. We guess that around 10 million words we will see the Zipfian tail and have a full understanding of morphosyntactic tag distribution for Turkish. Much of that, we expect, will be rare usages, therefore linguistically more interesting. (Manning (2011) arrives at similar conclusions.) Experiment 45, which repeats Experiment 41 with 5 million words, shows that almost all statistically significant uses have been covered.

5 Conclusion

The results of morphological analyzers and disambiguators seem to provide a rich base on which words can be processed without morphological analysis. In this work we show how they serve morphosyntactic tagging in this manner. We refer to them as morphological supertags because they summarize the morphological structure of a language succinctly. The choice of the tagset from this base seems to depend on the information needs of a particular application. The most comprehensive ones, i.e. parsers which also deliver some kind of meaning representation, require a tagset size of around a thousand, which coincides with our findings and with results reported for other languages. Some applications can perhaps make do with 20 tags, and some with 1,500. The full set, which is in our case 5,917, appears to be of some interest to linguists looking at the fine structure of morphosyntax. Our results show that tagset size and tagging accuracy are not linearly dependent, thus tagset choice is crucial. Choosing them on the basis of frequency alone does not scale up nicely, at least for the datasets that we used.

Acknowledgments

This research is partially supported under the grant 710477 of The Scientific and Technological Research Council of Turkey. The authors would like to thank Deniz Yüret for providing the corpus, Kemal Oflazer for providing the morphological analyzer, Haşim Sak for providing the morphological disambiguator and Mehmet Tatlıcıoğlu for his comments.

References

Thorsten Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLP, pages 224–231, Seattle, Washington, USA, April. ACL.

Ruken Çakıcı. 2008. *Wide-coverage Parsing for Turkish*. Ph.D. thesis, University of Edinburgh.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP, pages 1–8, Stroudsburg, PA, USA. ACL.

Bekir Taner Dinçer, Bahar Karaoğlan, and Tarık Kışla. 2008. A Suffix Based Part-of-Speech Tagger for Turkish. In *Fifth International Conference on Information Technology: New Generations*, ITNG, pages 680–685, Las Vegas, Nevada, USA. IEEE Computer Society.

Saso Dzeroski, Tomaz Erjavec, and Jakub Zavrel. 2000. Morphosyntactic Tagging of Slovene: Evaluating Taggers and Tagsets. In *Proceedings of the Second International Conference on Language Resources and Evaluation: LREC 2000*, pages 1099–1104.

David Elworthy. 1995. Tagset Design and Inflected Languages. In *Proceedings of the ACLSIGDAT Workshop*, pages 1–10.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of Turkish. *Computational Linguistics*, 34(3):357–389.

Aslı Göksel and Celia Kerslake. 2005. *Turkish: A Comprehensive Grammar*. Routledge. Third reprint, 2010.

Jan Hajic. 2000. Morphological Tagging: Data vs. Dictionaries. In *Proceedings of the Sixth Applied Natural Language Processing Conference: ANLP 2000*, pages 94–101.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical Morphological Disambiguation for Agglutinative Languages. *Computers and the Humanities*, 36:381–410.

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87.

Jorge Hankamer. 1989. Morphological parsing and the lexicon. In W. Marslen-Wilson, editor, *Lexical Representation and Process*. MIT Press, Cambridge, MA.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):356–396.

Selçuk Köprü. 2011. An Efficient Part-of-Speech Tagger for Arabic. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing*, volume 1 of *CICLing'11*, pages 202–213. Springer-Verlag.

Christopher D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is it Time for Some Linguistics? In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing*, volume 1 of *CICLing'11*, pages 171–189. Springer-Verlag.

Kemal Oflazer and Gökhan Tür. 1997. Morphological Disambiguation by Voting Constraints. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 222–229, Madrid, Spain, July. ACL.

Kemal Oflazer and İlker Kuruöz. 1994. Tagging and Morphological Disambiguation of Turkish Text. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 144–149, Stuttgart, Germany, October. ACL.

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür, 2003. *Building A Turkish Treebank*, pages 261—-277. Kluwer Academic Publishers.

Kemal Oflazer. 2003. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 29(4):515–544.

Csaba Oravecz and Peter Dienes. 2002. Efficient Stochastic Part-of-Speech Tagging for Hungarian. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, LREC, pages 710—717, Las Palmas.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological Disambiguation of Turkish Text with Perceptron Algorithm. In *CICLing 2007*, volume LNCS 4394, pages 107–118.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2011. Resources for Turkish Morphological Processing. *Language Resources and Evaluation*, 45:249–261.

Dan Tufiş and Oliver Mason. 1998. Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation*, LREC, pages 589–596.

Tamas Varadi and Csaba Oravecz. 1999. Morphosyntactic ambiguity and tagset design for Hungarian. In *Proceedings of the EACL LINC Workshop on Annotated Corpora*, Norway.

Deniz Yüret and Ferhan Türe. 2006. Learning Morphological Disambiguation Rules for Turkish. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 328–334, New York City, USA, June. ACL.