Morphology by Itself and Morphology for Itself: Morpho-supertagging without analysis

Abstract

In this paper we propose to use accumulated morphological knowledge of a language as a resource to do computational morphology without morphological analysis. We offer morphological supertags and language morpho-modeling as its basic mechanisms. The method consists of collecting large sets of morphological tags for a language, automatically disambiguating a corpus with these tags using an available method for the language, and building a language morpho-model using the supertag sets which guesses the most likely supertag in testing, if a word is not one of the morphologically uniquely supertagged in training. Our concerns in the paper are threefold: the feasibility of coming up with a supertag set for this task, the language morpho-model over supertags, and a method for assessment of supertag use. We show the application of our method to Turkish.

1 Introduction

Morphological tagging can be defined as the problem of finding a label for a word to represent the morphological properties of the word. Assuming that morphology is part of grammar, as it has been conceived since ancient times, these tags will include grammatical information as well. POS tagging is a kind of morphological 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 the basic parts of speech (POS).

The title of the paper borrows from that of Aronoff (1994), where morphology is conceived not as a means for something else (e.g. for syntax or phonology) but as a domain that has its own means and ends. The computational mechanism we add to this view is a statistical use of what we call *morphological supertags* for words, which are tags that relate to everything but the lexical content of the stem, and which give a complete description of a word's morphological properties (but not necessarily its morphotactics). We do not make *a priori* distinction between derivational and inflectional morphology, and rely on an analyzer's notion of stem.

The idea is adapted from supertagging in syntax (Bangalore and Joshi (1994; 1999)). We try to localize complexity of morphological structure in words by enumerating their morphological properties in training, in an attempt to simplify globally, i.e. computationally; see Bangalore and Joshi (2010) for the use of the idea in several grammar formalisms.

Our method for morphology is as follows. For a particular language, we collect most of its morphological tags reported in the field. We run this set through a state-of-the-art morphological disambiguator to obtain a set of semi-gold (supertag,word) pairings of a large corpus. Let us call this set M_g^l for language l. We look at test data: if a word in it uniquely matches a word in M_g^l, then we assume to have obtained the morphological supertag for the word, drawn from M_g^l. If not, a language morpho-model trained on M_g^l gives us the most likely supertag for the word in the test data. Our method's inputs are a set of disambiguated taggings and a language morpho-model trained solely on them, therefore it uses no component other than morphology to constrain morphology.

Morphological supertags are morphological properties (stem categories, affixes and morphological processes) made transparent. For example, the Turkish word *görmesinde* ‘his/her seeing’ receives the analysis *gör-me-si-nde* (see-V-INF-POSS3S-LOC) from the analyzers. The sequence -V-INF-POSS3S-LOC is its morphological supertag in M_g^t. It happens to encode morphotactics as well. For the Arabic word *duhhika* ‘made to laugh’, in which /h/ gemination from *dahaka* 'laugh' is the causative and /a/→/u/ is the passive for this verbal paradigm, we might receive *dahaka*-V-CAUS-PASS or *dahaka*-V-PASS-CAUS depending on the analyzer. The sequence starting with -V would be its supertag in M_g^a. What we expect to get from the analyzers during training is consistent delivery of results at the level of morphemes. We do not expect morphotactic information for all languages. We assume that morphological supertags as representatives of affixes and morphological processes can be abstracted away from the stem, to be used as the basis of our method. For example, Dreyer et al. (2008) and Dreyer and Eisner (2011) show how morphological processing for German can deliver ab**zu**brech*en* (different fonts indicate different morphemes) 'to be broken off' as OFF-TO-break-INF, or just *brech* and {OFF-,TO-, -INF}. Similarly for circumfixes such as *ge*rieb*en* ‘ground’ as *rieb* and {PASTPART}. Köprü (2011), Buckwalter (2004) and Habash et al. (2005) implement analyses of Arabic words to deliver results as exemplified above. Concatenative morphology has received due attention in the field as well.

Linguistic theorizing suggests that this is possible cross-linguistically. For example autosegmental morphology (McCarthy, 1981) transparently handles the delivery of morphemes whether they arise from items or processes, and whether they are contiguous (English), agglutinating (Turkish), discontinuous as in some German and Arabic morphology, or inflecting as in Latin and Czech. Bird and Ellison (1994) showed that the computational mechanism that underlies autosegmental (therefore serial or serializable) morphology is essentially finite-state. Such results testify for the availability of transparent methods for the delivery of morphemes.

Our working assumption is that if a word is morphologically ambiguous in the training data, in the sense of having multiple morphological bracketings or different tags in syntactic contexts, all of them will be represented in M_g^l. For example, the Turkish word *elle*, with stem *el* ‘hand’, is ambiguous in this sense; it can be *el*-INST ‘with hand’ (the inflectional reading of *-le*) or *el*-DER ‘touch’ (the derivational reading). Genuine ambiguities arise in contexts such as *elle karıştır* ‘mix with hand’ or ‘touch and stir’. Stems can be ambiguous too: *çizmeleri* is *çizme*[N]-POSS3P (*çizme-leri*) ‘their boots’, or *çiz*[V]-INF-PERS3P (*çiz-me-leri*) ‘their drawing’.

Ideally M_g^l would be a gold standard, decided by a native speaker. Short of this data on a large scale to do wide-coverage open-ended morphological processing, which is our main target in the long run, we use the semi-gold one as an approximation. It is semi-gold because we expect it to be semi-automatically disambiguated to suit our purpose of looking at large collections of earlier work.

The important point is that this set is disambiguated somehow, and this is the only place where the semantic aspects of morphology (its connection to a lexical or syntactic resource) come into play in our model. We think that this aspect also models the influential Separation Hypothesis (Beard, 1995), which amounts to saying that morphological types (stems, affixes, processes) cannot do semantics by themselves. This way we avoid the exponential search in e.g. two-level morphology that is involved in surface form to lexical form matching during morphological analysis (Koskenniemi, 1983; Koskenniemi and Church, 1988; Barton et al., 1987).

2 Morpho-supertags: is there a limit?

From a computational perspective, one crucial aspect of our way of thinking is the size of the supertag set in M_g^l. Two issues arise, (i) the practical aspects, i.e. performance, and (ii) theoretical aspects, e.g., whether (and how) languages manifest a manageable inventory of morphological differentials. For the practical aspect, if the supertag set is exhaustive, recall will surely increase as it will capture most of the distinctions in morphological use but at the expense of lower precision and/or difficulty of further experimentation. If the set is too small, it will be quite successful but it would suffer in further use e.g. in parsing, dependency detection and sentiment analysis. The theoretical aspect (ii) is put to test in

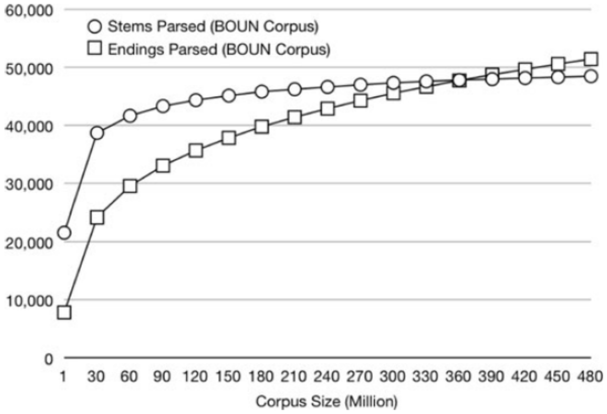

Figure 1: The number of newly encountered stems and supertags as functions of corpus size in T490, from Sak et al. (2011), Fig.3.

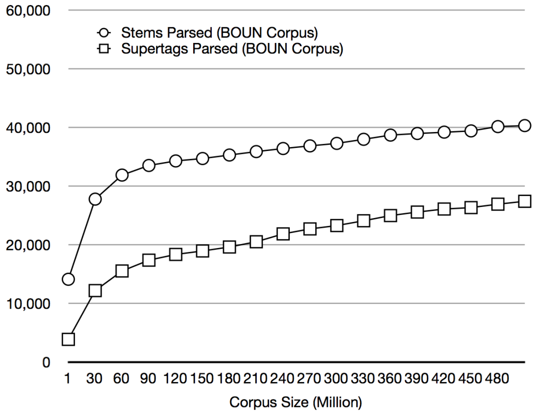

Figure 2: The number of newly encountered stems and supertags in BOUN/T490 using one feature per morpheme in a word.

languages like Turkish, where productive affixation and word-internal recursion seem to create an enormous number of possible affix/process sequencing.

Turkish facts arise from agglutination. There is no upper bound on suffixation. For example, the morpheme sequence "-de-ki" (-LOC-ki) can repeat any number of times—because of the relative marker *ki*—as long as it can find the right stem (Hankamer, 1989; Göksel and Kerslake, 2005): *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 a painting belonging to the family in the house.

Our definition of morphological supertags might avoid the scalability problem of open-ended corpora by not taking the stem as part of the supertags, or it might not. It might in principle suffer from recursive morphology. We eschew these aspects in M^l_g by taking into consideration only the morphemic supertags that occur in the training corpus.

How much information we lose on both practical and theoretical aspects by doing this can be evaluated if we have a large corpus. We suggest that doing the assessment with a language like Turkish and a very large corpus is essentially equivalent to discussing the idea asymptotically because of the sheer number of morphologically differentiable forms.

One such resource for Turkish is provided by Sak et al. (2011), which consists of 490 million words, hereafter referred to as BOUN corpus, following them, and as T490 when its size matters. Figure 1 shows their results. It appears that, after 480 millions words, we see more than 52,000 supertags (possible endings in the case of Turkish), which do not contain the stem. The numbers continue to increase as the corpus grows. They report 268 more endings for 490 million words. This sounds alarming for the idea of using early work as a collective resource to avoid morphological analysis.

A closer look at BOUN and its analysis by Sak et al. (2011) reveal that one reason for dispersion of information at possible endings might be the unusual number of features per supertag in a word. For example, the word (1a) receives the analysis (1b) in their semi-gold result.[1] There are only four morphemes in the word, as can be seen from (1a)'s gloss, but we see 18 morphemic features in (1b).

(1) a. çağdaş-laş-ma-cı-lık
contemporary-BECOME-INF-AGT-NESS
'modernism'

b. `çağdaş[Adj]-lAş[Verb+Become]+[Pos]`
`-mA[Noun+Inf2]+[A3sg]+[Pnon]+[Nom]`
`-CH[Noun+Agt]+[A3sg]+[Pnon]+[Nom]`
`-lHk[Noun+Ness]+[A3sg]+[Pnon]+[Nom]`

c. `çağdaş[Adj]-lAş-mA-CH-lHk`

When we look at BOUN from the perspective of morphemes, a trend similar to Figure 1 appears for stems in a smaller scale, reported in Figure 2, but

[1] The examples use the standard convention in Turkish to write morphemic descriptions of morphemes such as +SH and -mA, rather than their surface realizations depending on harmony and voicing, e.g. *sı*, *sü* for +SH. We kept the use of '-' and '+' in BOUN as different but consistent ways of marking morpheme labels.

Table 1: The BOUN/T490 corpus coverage with one feature per morpheme in a word.

Rank	Freq.	Cum.Per.	Morphemes
1	17,832,016	8.67	+SH
2	9,139,193	13.11	+NHn
3	9,067,575	17.52	+DA
4	8,169,109	21.49	+YA
1,000	5,755	97.80	-Hl-YHş+SH+NH
1,200	4,000	98.26	+mA+Hyor+YsA+nHz
5,000	139	99.83	-DHr-YAbil+Hyor+YsA
6,000	86	99.88	-t-YAmA-YAn+lAr+NHn
16,488	3	99.99	-YsA+YmHş
18,113	3	99.99	-Ar-DHk+lAr+HmHz+YlA
18,114	2	99.99	-YsA+k+DHr
20,972	2	100.00	-Ar-DHk+HnHz+NHn
20,973	1	100.00	-YsA+n+YsA
27,413	1	100.00	-Ar+DH+k

things look different for possible supertags. We disambiguated BOUN/T490 using the method of (Sak et al., 2011), which forms the basis of their results in Figure 1, and reduced the feature size of the supertag of every word to the number of morphemes in the word. We keep the morpheme label and avoid features like [Nom] and [Noun+Agt] in (1b), to transform (1b) to (1c). We repeated their experiment with the new supertag scheme. The result is in Figure 2. The supertag sets halved in number, from 52,000 plus of (Sak et al., 2011) to 27,413 in total, and also at almost every stage, while the stem list continues to grow as theirs.[2]

When we look at the number of supertags we need to provide a good coverage of T490, the results are encouraging for our method. Table 1 shows the rank and frequency of supertags in T490. These results are obtained after we reprocessed T490 with the idea of one-feature-per-morpheme as the algorithm for simplifying Sak et.al's supertags. We list the first and last rank of elements with frequencies less than four to give a feel for sparse use. If we discount single and double occurences, we reduce the supertags from 27,413 to 18,113, a further 34% reduction.

It seems that, around 1,200 tags, many languages' demand for tags begins to level off, to about 98% coverage. In the performance region of 90-98% accuracy, we have the following results. (We provide the sources of accuracies in §3.) Czech Treebank has 970 tags, Slovene 1,021, Hungarian 571, Romanian 611, and Turkish 1,350. (We reached the same accuracy for Turkish with 1,075 tags.) When tagging aims to aid parsing, a morphologically simple language needs tags in the order of hundreds or thousands as well. Clark and Curran (2007) use 425 tags for English that occur more than 10 times. Hockenmaier and Steedman (2007) use 1,286 lexical tags for wide-coverage parsing of English.

Also observe from Table 1 that, at around 5,000 supertags in T490, we reach a 99.83% morphological coverage of 490 million words. This number happens to coincide with another high-precision morphological analysis of Turkish, Yüret and Türe (2006), who use more than 4,000 tags to achieve 96% coverage. This is where supertags stop giving diminishing returns as well, because, as Table 1 shows, doubling or tripling an already overgrown supertag set could only give less than .2% in return, at the cost of making supertag set exploration a hopeless task.

Large tag or supertag sets complicate the tagging problem for morphologically rich languages. It is one of the reasons why most work attempts morphological disambiguation rather than morphological 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 supertag sets. Solving the morphological disambiguation problem, on the other hand, has a head start because it uses the output of a morphological analyzer to pick the correct morphological parse (Elworthy, 1995).

To answer our own question in this section's title: in theory, there is no cross-linguistic limit on the supertag set size. However, the idea of working with limited supertags while capturing most morphological distinctions *without* analysis *and* performing some potentially linguistically insightful exploration is computationally feasible. As Turkish is an extreme case manifesting recursive morphology, we think that our method and its exploratory mechanism can be of general interest.

We proceed as follows. Related work is covered in §3. Our four-stage method is described in §4. The experiments on choosing among the supertag sets

[2] Our stem counts also differ from theirs but the trend in this regard is similar. The difference might be due to the fact that we uncapitalized all the words, and left out numerals and words that are marked 'unknown' in their analysis.

are outlined in §5 and §6.

3 Related work

Our proposed method can be classified under morphosyntactic tagging, under which we also include POS tagging for reasons cited above. We know of no work that attempts to do it like ours, although the field is rich with supervised and unsupervised models of morphosyntactic/POS tagging.

Yüret and Türe (2006) perform tagging without morphological analysis. Their main focus is morphological disambiguation, not tagging, and they do not explore tag or supertag alternatives.

Most studies on disambiguation use their system as a tagger. However, they either use the output of a morphological analyzer in testing, or keep their tag set only at the level of basic part of speech tags. Oflazer and 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, to obtain 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 word-final tag, 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 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 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) 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 an ending-based POS tagger for Turkish (Dinçer et al., 2008), which fixes the number of graphemes of a word as 1..7 and apply Viterbi algorithm over simple HMMs. Their 90.2% success rate is best with a 5-gram. To give an idea on tag set sizes for methods based on analysis, Oflazer et al. (2003) use 1,350 tags.

For other morphologically rich languages such as Slovene, Czech, Hungarian and Romanian, we see work similar to the final stage of our work, which is selecting among proposed tags. Ours is a first attempt for Turkish. They have used full morphological descriptions and applied some techniques for reducing the tag set, such as removing the case, gender and number markers (Dzeroski et al., 2000; Varadi and Oravecz, 1999), or reducing the tag set which can be deterministically mapped to the full tag set (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. They report 96.59% accuracy for finding the basic POS tags with a tag set of size 12. Oravecz and Dienes (2002) also used the TnT tagger, for Hungarian, and measured 92.88% accuracy for full morphological tagging. Tufiş and Mason (1998) have built a reduced tag set, called C-tag set, with a size of 89 tags from a complete tag set size of 611 for Romanian. They call the method “Tiered Tagging”, which first finds a tag from the C-tag set 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 feature-based model for comparing tagging on several languages, with 92.96% accuracy for Czech. Another attempt at a universal tag set is Zhang et al. (2012), where language-particular tag sets are mapped by a model to a universal one with only 12 tags. The process improves the baseline performance of all the parsers tested for 19 languages. Our main concern is morphology without analysis, and we obtained 98.42% tagging accuracy in a similar experiment with 20 supertags only. However, such high accuracies are quite expected because of small target space, and they serve no useful function for exploring the morphologies of morphologically rich languages while avoiding analysis.

4 The language morpho-model (LMM)

Our overall method has a four-stage process. It consists of (i) collecting most of the morphological tags for a language in the form of supertags, (ii) automatically disambiguating a corpus with these supertags using an available method for the language, (iii) building a language morpho-model using the proposed supertag sets to return the most likely supertag sequence in testing (if a word is not one of the uniquely supertagged in training), and (iv) post-evaluating the supertags. It depends on analyzers for (i-ii), classifiers for (iii), and on some assessment and exploration tools for (iv).

Tagging is a classification problem as part of this scheme. The HMM is very successful in such problems, and easy to model. In principle, any classifier would do for our purpose, e.g. SVMs and perceptrons, as long as we can attempt to constrain morphology by morphology with supertag bigrams only.

Our language morpho-model is inspired by systems which employ the HMM approach to POS tagging in different languages, such as Brants (2000). We use the following formula:

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

The sequence of words in a string is represented as $w_1 \ldots w_n$. In order to calculate the estimate of the tag sequence $\hat{t}_1^n$, supertag transition probabilities and word likelihoods are used. The supertag transition probability $P(t_i|t_{i-1})$ is estimated with a bigram Language Morpho-Model (LMM) that is constructed from a supertag transition matrix, which is essentially a bigram of morphemic supertags including the stem's basic POS. 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 and the output symbols correspond to supertags. Viterbi algorithm is used to determine the most likely tag sequence as a supertag. With this approach we have captured stem-and-suffix morphology of Turkish as well as prefix-template/process-suffix morphology of Arabic.[3]

The critical role of LMM in our framework is its accuracy in predicting supertags for words which have not been seen in training. For this task, we obtained 94.20% accuracy of morphological tagging of Turkish, with 5,917 supertags for five million words. This result is higher than the only previous study on tagging of Turkish without a morphological analyzer (Yüret and Türe, 2006), which is 91.23% for a one million subset of our dataset. Recall also the 18,113 supertags for 490 million words of BOUN.

Clearly, having that many supertags might jeopardize the balance we might seek when we put these supertags to work for some task. They may provide too much information than can be useful, and using a small set of 20 provides too little to be of effective use. (We obtained 98.42% accuracy with 20 tags.) Very high accuracies at these extremes could therefore be misleading unless put in context.

The last part of our research concerns experiments in this regard. We considered several supertag set choices and their accuracies for Turkish. We introduce a novel use of the frequent-pattern tree for supertag set exploration, to automatically build a supertag set based on morpheme frequencies or their linguistic grouping.

5 The frequent-pattern tree

The Frequent Pattern-tree (Han et al., 2004), henceforth FP-tree, is a frequency tree of labels. It includes morpheme tags and the stem's POS tag in our case, but not the stem. Each node keeps a label and its frequency. We built our FP-trees for each dataset. For each tag of a morphological parse in training, we add a node to the tree, and increase its frequency in the order that the tag is situated in the parse result. Hence, each morphological tag sequence (supertag) corresponds to a path in the tree, and each frequency denotes the number of occurrences of a path in the corpus (for Turkish). For example, for the tags below, we build the FP-tree in Figure 3. Each node frequency is a total of its children's frequencies.

VERB-PAST-A1SG VERB-PAST VERB-NARR-A1SG NOUN
VERB-NARR-A1SG NOUN-PAST NOUN

FP-trees provide a condensed way of accumulating the morphological supertags. We can do experiments on the supertag set easily by manipulating a FP-tree by grouping its nodes, to understand the behavior of a language in a data-driven way. What we collect corresponds to the *history* of information in the corpus (McCallum, 1995), and we try to capture the statistical results of the history.

[3] Recall from introduction that this approach is suggested as a cross-linguistic framework if autosegmental morphology shows promise, that all morphologies are serial or serializable, with a finite-state computational base. In this regard, our method expects consistent delivery of results during training to establish supertag bigrams in a language.

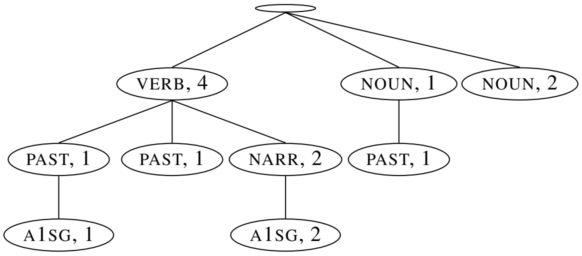

Figure 3: Sample tags and the corresponding FP-tree.

6 Supertag set alternatives for Turkish

We carried out two sets of supertag experiments. The first one is morphosyntactically motivated. The second set analyzes frequencies of supertags.

6.1 Unification of linguistic markers

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. In fact, the highest ranking supertag in 490 million words is a one-morpheme supertag, that of third person singular agreement usually labeled as +SH (Table 1).

Naturally, agreement tends to diffuse information *per word* because the dependency is not within words but among phrases.[4] We also know that, as an agglutinating language, the word-final tag must be crucial to its syntax. Some wide-coverage Turkish parsers make use of that information only (Oflazer, 2003; Eryiğit et al., 2008). Our results seem to provide an empirical justification of their choice.

Our main experiments regarding some linguistic grouping are listed in Table 2. Unification means treating a set of nodes as one in building the FP-tree frequency counts. The results we report in §6.2 refer to these experiments by their number.

6.2 Experiments

Our datasets and their use are as follows.

Table 2: Linguistic marker experiments.

Exp.	Experiment
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	Unifying 1ST, 2ND, 3RD person SINGULAR and PLURAL personal markers as PERSON
6	Unifying the markers specified in 2 and 3
7	Unifying the markers specified in 3 and 4
8	Unifying all markers
9	Unifying the markers specified in 2 to 4, and unifying the combined tags POSS and CASE markers as POSS-CASE, and PERSON and TENSE as TENSE-PERSON

T1 Approximately 1 million words of semi-automatically disambiguated dataset for Turkish, from Yüret and Türe (2006). It has 4,385 supertags. This is a commonly used dataset. It is considered semi-gold in Turkish studies. Its supertag set gives 99.52% morphological coverage of T490.

T5 Approximately 5 million words we have collected ourselves for training, and disambiguated. It includes T1. It has 5,917 supertags, which is a superset of that of T1. These 5,917 supertags give 99.69% morphological coverage of T490.

T490 The BOUN corpus of Turkish consisting of 490 million words (Sak et al., 2011). Their work extracted around 52,000 supertags, which we first reduced to around 18,000 by the processes described earlier. This corpus is used for verification and feasibility studies by us; it is not part of training. The supertag set of T5 is a subset of that of T490.

T5 is collected by us for bootstrapping. We morphologically analyzed it with Oflazer and Kuruöz (1994) parser, and disambiguated with the method of Sak et al. (2007). We use 96% of T1 for training, 2% for development and 2% for testing. Sak et al. (2007) use the same dataset as follows: 79% for training, 4.2% for development and 4.2% for testing. We use T5 for optimizing the supertag weights and the stems' POS weights in the language morphomodel. They are used for calculating the supertag probability of words that are not uniquely matched. We use the supertags of T1 and T5 for experimentation, and report T1 results unless noted otherwise. Their 99.5%+ morphological coverage of 490 million words corroborates for their adequacy.

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

Table 3: Unification of linguistic markers. Descriptions of the experiments are in Table 2.

Exp.	Supertag Set Size	Accuracy(%)
1	4,385	91.16
2	3,369	91.40
3	3,603	91.93
4	3,631	91.24
5	3,941	91.18
6	2,829	92.70
7	3,301	91.29
8	1,834	93.00
9	1,355	94.02

6.2.1 Method

For training, we first assume the morphological supertag of a word as its full morphological parse. Such results are available thanks to morphological analyzers and disambiguators. We use these supertags for the lookup of a test word, in matching a target pair from training (raw data × its supertag).

For example, the word *dünyasının* is morphologically *dünya-sı-nın*, analyzed as world-N-POSS3S-GEN3S 'of his/her world' for saving in M_g^t. The sequence beginning with -N is its morphological supertag (the remainder is considered to be the stem, in this case *dünya*). When the training set is large enough there is information about various wordforms of *dünya* (hence more supertags) to feed the LMM (§4). If a word uniquely matches *dünyasının* in testing, its supertag is found. If not, LMM provides the most likely supertag trained on T5.

We report accuracy results in FP-tree experiments. We built our FP-trees from T1 and T5.

6.2.2 Linguistic marker experiments

Table 3 lists the results of the linguistic experiments. The possessive marker manifests itself as the most efficient unified inflectional group (cf. first five rows of Table 3). The potential explanation for that was provided before (§6.1), which has to do with pervasiveness of Turkish agreement morphology. It mainly coincides with possessive marking. Recall that our data is semi-gold, not gold, hence it cannot fully distinguish nominal agreement morphology engendered by possessive marking from subordination's agreement morphology by more or less the same markers, both of which require syntax.

Table 4: Pruning the FP-tree based on averages. F:frequency of a node, avg_{tree}:average frequency of the tree, avg_n: average freq. based on most frequent n nodes.

Exp.	Pruning	Supertags	Accuracy
10	No pruning	4,385	91.16%
11	$F < avg_{tree}$	725	91.95%
12	$F < avg_{tree}/2$	917	91.65%
13	$F < avg_{tree}/4$	1,231	91.41%
14	$F < 2 \times avg_{tree}$	615	92.67%
15	$F < avg_{1,000}$	385	93.83%

Experiments 8 and 9 score the best accuracy results. This seems natural given the fact that they lump almost all Turkish inflection into one group (except voice and negation). Experiments 6–7 do the same in smaller groups. Experiment 6 crosscuts 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. Note also that the third and fourth most frequent supertags in T490 are one-morpheme supertags for locative and dative cases (Table 1). Experiment 6 unifies them with POSS.

6.2.3 Frequency experiments

We also conducted experiments on the effect of supertag frequency without a concern for the linguistic aspects. Table 4 describes the FP-pruning results. For example, Experiment 11 prunes nodes that are lower than average in frequency.

Pruning is different than unification: pruning at node x means discarding the children of x and keeping the count of x, rather than reorganizing the tree by recalculating the sums as done by unification.

The average frequency in the tree is calculated as 138 for T1. In these experiments, number 15 stands out. Its supertag set is probably too weak to serve a parser, but it can perform well for opinion mining or sentiment analysis where lighter annotation suffices.

In another set of experiments (Table 5), only the most frequent supertags are kept, and less frequent supertags are mapped to the frequent ones. The mapping is performed in the FP-tree by collapsing the child nodes to upper levels. The aim in these experiments is to observe the relation between the number of significant tags and accuracy. Checking by frequency, we found that the most frequent 1,000 tags cover 99.2% of T1. For comparison, Figure 4

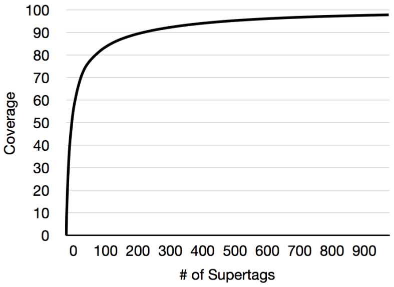

Figure 4: 1,000 most frequent supertags of T490.

Table 5: Pruning by most frequent *n* supertags.

Exp.	Most frequent	Accuracy
16	800	91.66%
17	1,000	91.66%
18	1,200	91.56%
19	2,000	91.49%

shows the same result for 490 million words.

Table 5 shows the results for T1. Note that the experiments reported in Table 4 and Table 5 are different. For example, in Experiment 15 we remove all nodes with frequency below the average of most frequent 1,000 tags. Therefore, the supertag set generated in Experiment 15 is covered by the supertag set generated in Experiment 17. Based on their accuracy, we would choose the smaller supertag set (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 that apply to the stem is 7. In the experiments 20 to 27, the FP-tree is pruned by levels. The aim in these experiments is to observe the effect of the supertag length in morphological tagging. For example, in Experiment 20, only the POS of the stem is taken and all remaining morphemes are discarded because level 1 in the FP-tree contains the basic POS tags only. (Recall that we report T1 results by default; 20 supertags reach 98.42% accuracy in T5.)

Pruning the FP-tree by supertag length did not perform well; cf. the lower accuracy rates in Table 6. For example, in Experiment 21, the first 2 levels of the FP-tree are used and the supertag set size is only 171. However, the accuracy is 93.84%, which is a poor performance compared to Experiment 9 with 1,355 supertags and a comparable 94.02% accuracy.

Table 6: Pruning FP-tree according to supertag length.

Exp.	Pruning level	Supertags	Accuracy
20	1 (keep stem's POS)	20	97.21%
21	2	171	93.84%
22	3	998	91.56%
23	4	2,621	91.53%
24	5	3,886	91.22%
25	6	4,290	91.18%
26	7	4,376	91.16%
27	8 (no pruning)	4,385	91.16%

In the last set of pruning experiments, we explore the relation between morpheme position and morphological tagging. For this aspect we make special use of the morphological analyzers for Turkish as we did for the previous experiment set: they provide morpheme ordering as well as morpheme tagging. The supertag sets 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 7. In Experiment 28, we observed the most robust tagging performance when the last morpheme in the supertag is removed (94.40% with 1,440 supertags), which is about the same supertag set size as in other languages for that level of accuracy. It seems that the last morpheme is not for the benefit of supertag performance; it is for syntax. 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 the least dependent one on the word itself. Its performance can be appreciated better when this result is compared with Experiment 18 (1,200 supertags and 91.56% accuracy), or with Experiment 21 (171 supertags and 93.84% accuracy). The closest accuracy to Experiment 28 at this supertag set size is achieved in Experiment 9 (1,355 supertags 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 supertag set at least doubles in size. It seems to suggest that we must keep agreement information for parsing, rather than live with supertag set sizes around 3,248 and gain no significant advantage in other tasks such as sentiment analysis etc.

Experiments 33-39 need explaining. 33 keeps the basic POS tag of the stem and the last morpheme tag

Table 7: The effect of the final morphemes in a supertag. POS is the first tag in a supertag.

Exp.	Adjustment	Supertags	Accuracy
28	Prune final morpheme	1,440	94.40%
29	Prune final morpheme if agr.	3,248	91.49%
30	Prune all and only agreement	2,804	91.79%
31	Prune last 2 morphemes	498	95.69%
32	Prune last 3 morphemes	183	96.52%
33	Keep POS & last morpheme	273	94.00%
34	Keep POS & last 2 morphemes	1,418	91.83%
35	Keep POS & last 3 morphemes	2,948	91.31%
36	Keep POS & last 4 morphemes	3,977	91.22%
37	Keep POS & last 5 morphemes	4,309	91.19%
38	Keep POS & last 6 morphemes	4,379	91.17%
39	Keep POS & last 7 morphemes	4,385	91.16%

of any word in training, e.g. -N-GEN3S for *dünya-sı-nın*, which has the supertag -N-POSS3S-GEN3S ('of his/her world'). Given its relatively good success with a moderately discriminating supertag set of 273, it seems good enough for cases where lexical semantics may also be added from stems, as in polarity items (likewise #31). Finally, we removed the single/double occurrences of supertags, which did not improve the results much compared to the baseline. From this result we conclude that the training data does not contain much noise. We also repeated the full supertag set experiment (#39) with T5. We gain 3.4% more in accuracy to reach 94.20%, with 1,532 more supertags (from 4,385 to 5,917).

7 Discussion and conclusion

We suggest that computational morphology in its current state has enough resources to constrain itself for training our models, to use them during performance without analysis. Our method depends on large sets of morpheme labels to do morphology without reference to a lexical or syntactic resource in testing, i.e. to confine complex information to training in order to simplify computation.

We call these resources morphological supertags. We think they suffice to build a language morpho-model to estimate the supertags of words. Large-scale semi-gold data is needed for this way of thinking, and it is affordable now because of rich morphological work on many languages.

We hope to have shown the feasibility of various supertag sets as well. The choice of supertags depends on the information needs of a particular application. The most comprehensive ones, which are parsers which deliver a meaning, require a supertag set size in the order of a thousand, which coincides with our findings and with the results reported for other languages (e.g. 1,286 tags for wide-coverage of English reported earlier).

For Turkish morphology in particular, we reached an accuracy of 94.20% with 5,917 supertags, and 98.42% with 20 POS tags. To our knowledge, these are the highest results on full morphological tagging of Turkish without the use of a morphological analyzer in testing, which are comparable with the best result that uses a morphological analyzer for training and testing (96%).

Having a large set requires an exploration, and current work is a first attempt for Turkish. Hakkani-Tür et al. (2002):386 considered this task to be infeasible, that there can be no simple tag set design for Turkish because of the syntactic possibilities involved. Their case in point is examples like *masamdakiler* 'those (things) on my table' (masa-N-POSS1S-LOC-KI-PLU), which, according to them, could be a noun (because the root and the final inflection is nominal), or adjective, because *-ki* can derive adjectives in Turkish. This use of *-ki*, however, is not adjectival because of the post inflection after it: **masamdakiler kalem*, (*kalem* for 'pencil'), which is ungrammatical as an adjective but fine as a pronominal 'those (things) on my table are pencils'. Compare *masamdaki kalem* 'the pencil on my table'. Only syntax can tell us which *-ki* to use, and a morphemic way of doing syntax relieves the burden from morphology by not asking it to provide a lexical resource or one overall POS result per word (Bozsahin, 2002). Consider also what proactive morphology could do for word forms such as *about's* and *know's*, in: *the brother I was telling you about's taste in wallpaper*, or, *the man I know's taste in wallpaper* (Anderson et al., 2006). Morphological supertags without analysis avoids such problems, and in this sense they are similar in spirit to supertagging in syntax. Hakkani-Tür et al.'s concern arises when there must be some morphological analysis (to the extent of reaching for a lexical resource) before syntactic processing begins. This is known to complicate both parsing and morphological processing, as noted in the introduction.

Our datasets and code are publicly available at our web site.

References

Stephen R. Anderson, Lea Brown, Alice Gaby, and Jacqueline Lecarme. 2006. Life on the edge: there's morphology there after all! *Lingue e Linguaggio*, 1(1):33–47.

Mark Aronoff. 1994. *Morphology by Itself: Stems and Inflectional Classes*. MIT Press, Cambridge, MA.

Srinivas Bangalore and Aravind K. Joshi. 1994. Disambiguation of super parts of speech (or supertags): Almost parsing. In *Proceedings of the International Conference on Computational Linguistics (COLING 94), Kyoto University, Japan, August*, San Francisco, CA. Morgan Kaufmann.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25:237–265.

Srinivas Bangalore and Aravind K. Joshi, editors. 2010. *Supertagging*. MIT Press, Cambridge, MA.

G.E. Barton, R.C. Berwick, and E.S. Ristad. 1987. *Computational Complexity and Natural Language*. MIT Press, Cambridge, MA.

Robert Beard. 1995. *Lexeme-Morpheme Base Morphology*. SUNY Press, Albany, NY.

Steven Bird and T. Mark Ellison. 1994. One-level phonology: Autosegmental representations and rules as finite automata. *Computational Linguistics*, 20(1):55–90.

Cem Bozsahin. 2002. The combinatory morphemic lexicon. *Computational Linguistics*, 28(2):145–176.

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.

Timothy Buckwalter. 2004. Buckwalter Arabic morphological analyzer, version 2.0. Linguistic Data Consortium.

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.

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 616–627, Stroudsburg, PA, USA. ACL.

Markus Dreyer, Jason R. Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1080–1089, Stroudsburg, PA, USA. ACL.

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.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan, June. Association for Computational Linguistics.

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.

Kimmo Koskenniemi and Kenneth Ward Church. 1988. Complexity, Two-level morphology and Finnish. In *Proceedings of COLING*, pages 335–339.

Kimmo Koskenniemi. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, Department of General Linguistics, University of Helsinki.

R. Andrew McCallum. 1995. Instance-based utile distinctions for reinforcement learning with hidden state. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 387–395. Morgan Kaufmann.

John J. McCarthy. 1981. A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry*, 12(3):373–418.

Kemal Oflazer and Ilker 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 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, 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. Morpho-syntactic 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.

Yuan Zhang, Roi Reichart, Regina Barzilay, and Amir Globerson. 2012. Learning to map into a universal POS tagset. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1368–1378, Stroudsburg, PA, USA. ACL.